cpmPlus™

Loop Performance Manager 3.3

User Manual

This software is licensed by ABB. General Terms and
Conditions of Sale only for use on this workstation.
© Copyright 2003, ABB. All rights reserved.
ABB

ABB. All rights reserved

NOTICE

The information in this document is subject to change without notice and should not be construed, as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this document.

In no event shall ABB be liable for direct, indirect, special, incidental or consequential damages of any nature of kind arising from the use of this document nor shall ABB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and part thereof must not be reproduced or copied without ABB's written permission and the content thereof must not be imparted to a third party nor be used for any unauthorized purpose.

The software described in this document is furnished under a license and may only be used, copied or disclosed in accordance with the terms of such license.

TRADEMARKS

Registrations and trademarks used in this document include:

ActiveX	Registered trademark of Microsoft Corporation.
Honeywell	Registered trademark of Honeywell Corporation.
Java	Trademark of Sun Microsystems, Inc.
MATLAB	Registered trademark of The Math Works, Inc.
Metso, Metso DNA	Registered trademark of Metso
Microsoft	Registered trademark of Microsoft Corporation.
OPC	Registered trademark of OPC Foundation.
Pentium	Registered trademark of Intel Corporation.
Siemens, Siemens Teleperm	Registered trademark of Siemens
Windows	Registered trademark of Microsoft Corporation.

Any other company and product names used herein may be the trademarks or the registered trademarks of their respective companies.

Copyright © 2003-2010 ABB
All rights reserved.

Table of Contents

1. OVERVIEW 13

1.1 About This Manual 13
1.2 Who Should Read This Manual 13
1.3 Prerequisites 13
1.4 Document Conventions 14
1.4.1 Graphical User Interface Elements 14
1.4.2 Note or Warning 14
1.5 Terminology 14
1.6 How This Manual Is Organized 15

2. INTRODUCTION 16

2.1 Product Overview 16
2.2 Prerequisites and requirements 17
2.2.1 Workstation Hardware 17
2.2.2 Connectivity – OPC case 17
2.2.3 Connectivity – Infi90 case 17
2.2.4 Connectivity – KM 6.1 case 18
2.3 Intended user 18
2.4 Product description 18
2.4.1 Loop Auditing 18
2.4.1.1 Auditing Status Monitor 20
2.4.2 Control Loop Tuning 20
2.4.2.1 Data collection 21
2.4.2.2 Model identification 22
2.4.2.3 PID Tuning 27
2.4.2.4 Feedback Tuning rules 33
2.4.2.5 Feedforward Tuning 36
2.4.2.6 Tuning Cascade Loops 39
2.4.3 Web Client 42
2.4.3.1 What Auditing means to you 43

3. INSTALLING LPM 44

3.1 How to install LPM 44
3.2 Installation Summary 45
3.3 Post installation instructions 45
3.4 Software Licensing 52
3.4.1 License Entry Program 52
3.4.1.1 Requirements 52
3.4.1.2 Install 52
3.4.1.3 Uninstall 52
3.4.1.4 Running the License Entry Program 53
3.4.2 Selecting a Machine ID 53
3.4.3 Requesting Software Keys 54
3.4.4 Entering Licenses 55
3.5 Startup and Shutdown Procedures 57
3.6 Validation string 57

4. CONFIGURATION/APPLICATION BUILDING 58

4.1 Auditing Workflow 58
4.2 Tuning Workflow 59
4.3 LPM Client Tool Overview 60
4.4 Toolbar 61
4.5 Configuration Tree 63

ABB. All rights reserved

5. TUNING 66

5.1 CONFIGURATION 66
5.1.1 Area Configuration 66
5.1.1.1 OPC Data Source Type Configuration 67
5.1.1.2 Infi90 Data Source Type Configuration 67
5.1.1.3 Knowledge Manager 6.1 / cpmPlus History 69
5.1.2 Loop Configuration 69
5.1.2.1 Manual Entry 70
5.1.2.2 Bulk Data Configuration Utility 75
5.1.2.3 Excel File Required Format 79
5.2 DATA COLLECTION 81
5.2.1 Importing Data from MS-Excel® files 83
5.3 MODELING 87
5.3.1 Identification Tab 89
5.3.1.1 Collected Data elaboration 89
5.3.1.2 Basic Mode 92
5.3.1.3 Advanced Mode 93
5.3.1.4 Defined Model assessment 95
5.3.2 Model Response Tab 98
5.4 CONTROLLER TUNING 101
5.4.1 FB Tune and Simulate Tab 104
5.4.1.1 Manual Tuning 109
5.4.1.2 Lambda Tuning 111
5.4.1.3 Dominant Pole Placement PI or PID 111
5.4.1.4 IMC tuning PI and PID 112
5.4.1.5 Lambda-Smith 113
5.4.1.6 Integral Time Absolute Error (ITAE) 113
5.4.2 FB Evaluation Tab 117
5.4.3 FF Tune and Simulate Tab 119
5.4.4 FF Evaluation Tab 120
5.5 TUNING LOGS 122
5.6 CASCADE CONTROL LOOPS 123

6. AUDITING 128

6.1 INTRODUCTION 128
6.1.1 Basic Nomenclature 128
6.1.2 Summary 129
6.2 CONFIGURATION 132
6.2.1 Plant Areas 134
6.2.2 Loops 134
6.2.3 Loop Categories 139
6.2.3.1 Batch Category/Shifts Résumé 141
6.3 ASSESSING LOOP PERFORMANCES 144
6.3.1 Performance Indices 144
6.3.2 Diagnoses 145
6.3.3 Plotting Indices and Diagnoses 146
6.4 REPORTS 150
6.4.1 Report Configuration 151
6.4.2 Area Tag Reports 154
6.4.3 Report Generation 155
6.4.4 Loop Plots 160
6.5 INTERACTIVE REPORTS 162
6.5.1 Loops report 162
6.5.2 Aggregate report 162
6.5.2.1 Area Overview Tab 163
6.6 ADVANCED AUDITING SETTINGS 163
6.6.1 Prediction Horizon Tab 164
6.6.2 Threshold Matrix Tab 164

ABB. All rights reserved

6.6.3 Rank and Rank score concepts ... *165*
6.6.4 Performance Weights Tab ... *166*
6.6.5 Performance Levels Tab ... *166*
6.7 MONITOR SERVER STATUS ... 167
6.8 LPM WEB CLIENT ... 170
6.8.1 LPM Web Client Installation ... *178*

7. LPM INTERFACE WITH KM 6.1 ... 188

7.1 OVERVIEW ... 188
7.2 SCOPE AND REQUIREMENTS ... 188
7.3 CONFIGURING THE KM SERVER ... 189
7.3.1 Test Connection to KM Server ... *189*
7.4 KM AS TUNING DATA SOURCE ... 190
7.4.1 Configure KM Area ... *190*
7.4.2 Configure KM Loop ... *190*
7.4.3 Data Collection ... *191*
7.5 KM AS AUDITING DATA SOURCE ... 194
7.5.1 Configuration ... *194*
7.5.2 Raw audit data collection ... *194*
7.5.3 Audit data collection ... *195*
7.6 KM AS KPI DATABASE ... 196
7.7 LPM REPORTS IN KM PERSONAL ASSISTANT ... 197
7.7.1 Standard LPM reports in KM ... *197*
7.7.2 Importing Standard Reports in KM ... *197*
7.7.3 Viewing the LPM reports in the KM personal Assistant ... *197*
7.7.4 System Reports ... *198*
7.7.4.1 Overview Report ... 198
7.7.4.2 Inactive Indices Report ... 198
7.7.5 User Reports ... *199*
7.7.5.1 Loop Audit Data and Loop Audit Data Averaged ... 199
7.7.5.2 Loop Audit Diagnostics ... 200
7.7.5.3 Loop Audit Preconditions ... 200
7.7.5.4 Loop Audit Diagnostics Best Worst – Worst Best ... 200

8. UTILITIES ... 202

8.1 LOOP FIND ... 202
8.2 CONFIGURATION SUMMARY ... 202
8.2.1 General tab ... *203*
8.2.2 Area Allocation ... *204*
8.2.3 Category Allocation Tab ... *204*
8.2.4 Controller type ... *205*
8.3 OPTIONS ... 205

9. APPENDIX A - CHART OPERATING INSTRUCTIONS ... 208

10. APPENDIX B – TRANSFER FUNCTIONS ... 211

11. APPENDIX C – TECHNICAL DETAILS ... 217

11.1 TUNING EVALUATION PARAMETERS ... 217
11.1.1 Time response quantification ... *217*
11.1.1.1 Overshoot ... 217
11.1.1.2 Rise time ... 217
11.1.1.3 Settling time ... 217
11.1.1.4 OP overshoot ... 217
11.1.1.5 Maximum deviation ... 218
11.1.1.6 IAE ... 218
11.1.2 Frequency response quantification ... *218*
11.1.2.1 Gain margin ... 218

ABB. All rights reserved

11.1.2.2 Phase margin 218
11.1.2.3 Delay margin 218
11.1.2.4 Maximum sensitivity 219
11.1.2.5 HF noise gain 219
11.1.2.6 Bandwidth 219
11.1.2.7 LF asymptote 219
11.1.2.8 LF slope 220

12. APPENDIX D - DESCRIPTION OF THE PERFORMANCE INDICES 221

12.1 MEAN LOOP ERROR 222
12.2 STANDARD DEVIATION LOOP ERROR (NORMALIZED) 223
12.3 STANDARD DEVIATION LOOP ERROR (ENG. UNIT) 224
12.4 MEAN PROCESS VARIABLE 225
12.5 STANDARD DEVIATION CONTROLLER OUTPUT 226
12.6 SKEWNESS CONTROL ERROR 227
12.7 KURTOSIS CONTROL ERROR 228
12.8 DATA COMPRESSION INDEX 229
12.9 DATA QUANTIZATION INDEX 230
12.10 OUTLIER INDEX 231
12.11 NOISE LEVEL INDEX 232
12.12 RATIO CONTROL ERROR VS. CONTROLLER OUTPUT INDEX 233
12.13 AUTOMATIC MODE INDEX 234
12.14 SATURATION INDEX 235
12.15 CASCADE MODE INDEX 236
12.16 AUTO-CORRELATION RATIO 237
12.17 SETPOINT CROSSING INDEX 238
12.18 CASCADE TRACKING INDEX 239
12.19 CASCADE TRAVEL RATIO INDEX 240
12.20 HARRIS PERFORMANCE INDEX 241
12.21 OSCILLATION INDEX ON CONTROL ERROR 242
12.22 OSCILLATION INDEX ON SETPOINT 243
12.23 OSCILLATION PERIOD 244
12.24 OSCILLATION AMPLITUDE 245
12.25 OSCILLATION SEVERITY INDEX 246
12.26 OSCILLATION AMPLITUDE REGULARITY INDEX 247
12.27 SHUT-OFF PROCESS VALUE INDEX 248
12.28 STICTION INDEX 1 (CROSS CORRELATION METHOD) 249
12.29 STICTION INDEX 2 (HISTOGRAM METHOD) 250
12.30 VALVE TRAVEL PER HOUR 251
12.31 VALVE REVERSAL PER HOUR 252
12.32 VALVE SIZING INDEX 253
12.33 NON-LINEARITY 1ST INDEX 254
12.34 NON-LINEARITY 2ND INDEX 255
12.35 NON-GAUSSIANITY INDEX 256
12.36 OSCILLATION SYMMETRY INDEX 257
12.37 ESTIMATED PROCESS TIME-DELAY 258
12.38 MASTER FLAG 259
12.39 SETPOINT STEP FLAG 260
12.40 MODEL FIT FLAG 261
12.41 STATIC GAIN INDEX 262
12.42 PREDICTION HORIZON 263
12.43 LOOP CATEGORY 264
12.44 AREA TAG 265
12.45 USER DEFINED INDICES 266
12.46 AUTOMATIC MODE INDEX (CONTINUOUS) 267
12.47 MEAN ABSOLUTE LOOP ERROR (CONTINUOUS) 268
12.48 SATURATED PERCENT (CONTINUOUS) 269
12.49 SUMMARY TABLE 270

ABB. All rights reserved

13. APPENDIX E - DESCRIPTION OF THE DIAGNOSIS HYPOTHESIS 275

13.1 ACCEPTABLE OVERALL PERFORMANCE 275
13.2 TUNING PROBLEM 276
13.3 LOOP OSCILLATORY 277
13.4 SETPOINT OSCILLATORY 278
13.5 SIGNIFICANT EXTERNAL DISTURBANCE 279
13.6 SIGNIFICANT NON LINEARITY 280
13.7 VALVE STICTION 281
13.8 VALVE LEAKAGE 282
13.9 VALVE SIZE INCORRECT 283
13.10 EXCESSIVE VALVE ACTION 284
13.11 DATA UNRELIABLE 285

14. APPENDIX F: PID CONTROLLER TYPES 286

14.1 BACKGROUND 286
14.2 GENERIC PID FORMS 287
14.2.1 PARALLEL 287
14.2.2 IDEAL 287
14.2.3 SERIES 287
14.3 INDUSTRIAL PID FORMS 288
14.3.1 ABB AC800 Series 288
14.3.1.1 AC800 F 288
14.3.1.2 AC800 M/C 288
14.3.2 ABB Bailey 289
14.3.2.1 FC18 and FC19 289
14.3.2.2 FC156 289
14.3.2.3 FC160 – Inferential Smith Controller 290
14.3.2.4 ABB Hartmann and Braun Contronic P 290
14.3.2.5 ABB CONTRONIC P PID 290
14.3.2.6 ABB Melody APID 291
14.3.2.7 ABB Advant/Master 291
14.3.2.8 ABB Mod300 292
14.3.2.9 ABB AlfaLaval Control 292
14.3.3 Honeywell TDC 293
14.3.4 Metso DNA 293

15. APPENDIX G: TAGLIST CREATOR, INFI90 294

15.1 INTRODUCTION 294
15.2 HOW TO IDENTIFY INFI90 LOOPS 294
15.2.1 First step: identify names and base address for loops 294

16. APPENDIX H: REGISTRY PARAMETER SETTINGS 298

16.1 OVERVIEW 298
16.2 LPM KEYS PATH 298
16.3 LPM KEYS DETAILS 299

ABB. All rights reserved

LIST OF FIGURES:

FIGURE 1 - AUDIT PLOTS 19
FIGURE 2 - REPORT CONFIGURATION 20
FIGURE 3 - DATA COLLECTION 22
FIGURE 4 - MODEL IDENTIFICATION: THE BASIC FORM 23
FIGURE 5 - MODEL IDENTIFICATION: THE ADVANCED FORM 24
FIGURE 6 - REMOVING BAD DATA FROM THE IDENTIFICATION DATASET 25
FIGURE 7 – PID TUNING 27
FIGURE 8 - MANUAL TUNING IN LPM 28
FIGURE 9 – TUNING METHOD GUIDELINES 31
FIGURE 10 – TEST PERFORMANCES AGAINST A MODEL 32
FIGURE 11 – CHANGE THE SIMULATION DURATION 33
FIGURE 12 - BLOCK DIAGRAM OF A PID CONTROL LOOP 34
FIGURE 13 - BLOCK DIAGRAM OF AN IMC LOOP 35
FIGURE 14 - FEEDFORWARD CONTROLLER 36
FIGURE 15 - FF TUNE & SIMULATE 38
FIGURE 16 - FF SIMULATE DETAILS 38
FIGURE 17 – CASCADE CONTROLLER 40
FIGURE 18 - CONFIGURATION FORM FOR CASCADE MASTER / OUTER LOOP 41
FIGURE 19 - LPM WEB CLIENT 42
FIGURE 20 - DIALOG BOX TO INSTALL LPM 44
FIGURE 21 - LICENSING AGREEMENT ACCEPT DIALOG 44
FIGURE 22 - DCOMCNFG DIALOGUE 46
FIGURE 23 - DCOM CONFIGURATION 46
FIGURE 24 - DCOM CONFIGURATION 47
FIGURE 25 - DCOM CONFIGURATION 48
FIGURE 26 - DCOM CONFIGURATION 49
FIGURE 27 - DCOM CONFIGURATION 50
FIGURE 28 - SECURITY TAB 51
FIGURE 29 - USER IDENTITY 51
FIGURE 30 - SOFTWARE LICENSING PROCEDURE 52
FIGURE 31 - AVAILABLE IDS TAB 53
FIGURE 32 - LICENSES TAB 54
FIGURE 33 -EDIT LICENSE TAB 56
FIGURE 34 – ENTERING THE VALIDATION STRING 57
FIGURE 35 – LOOP AUDITING WORKFLOW 58
FIGURE 36 – LOOP TUNING WORKFLOW 59
FIGURE 37 - MENUS AND MENU ITEMS 60
FIGURE 38 - TUNING TREE STRUCTURE 63
FIGURE 39 - MAIN BRANCHES IN THE AUDITING TAB 64
FIGURE 40 - AREA POP-UP MENU 66
FIGURE 41 - AREA CONFIGURATION FORM 67
FIGURE 42 - AREA CONFIGURATION FORM FOR INFI90 CONNECTION 68
FIGURE 43 – COMMUNICATION INTERFACE SETTING 69
FIGURE 44 - ADD LOOP 70
FIGURE 45 - ADD LOOP DIALOG 70
FIGURE 46 - LOOP TREE VIEW STRUCTURE 71
FIGURE 47 - LOOP CONFIGURATION FORM 72
FIGURE 48 - LOOP CATEGORY DROP-DOWN MENU 72
FIGURE 49 - PID CONTROLLER TYPE DROP-DOWN MENU 73
FIGURE 50 - GAIN MULTIPLIER FOR BAILEY PIDS 74

ABB. All rights reserved

FIGURE 51 - AUXILIARY TAGS TAB 74
FIGURE 52 - CONTROLLER CONSTANTS TAB 75
FIGURE 53 - LAUNCHING THE BULK DATA CONFIGURATION UTILITY 76
FIGURE 54 -THE BULK DATA CONFIGURATION UTILITY WINDOW 76
FIGURE 55 - THE FILE SELECTION DIALOG 77
FIGURE 56 - WARNING MESSAGES DURING TAG CONFIGURATION BUILD-UP 78
FIGURE 57 - BULK DATA CONFIGURATION RESULT 79
FIGURE 58 - DATA POP-UP MENU 81
FIGURE 59 - DATA COLLECTION WINDOW 82
FIGURE 60 - DATA COLLECTION EXAMPLE 83
FIGURE 61 – EXCEL IMPORT: DATASET NAME SELECTION 84
FIGURE 62 – EXCEL DATA IMPORT FORM 84
FIGURE 63 – EXCEL IMPORTED DATASET 85
FIGURE 64 – DISPLAYING EXCEL-IMPORTED DATASETS 85
FIGURE 65 - ADD MODEL FORM 87
FIGURE 66 - MODEL OPERATIONS 88
FIGURE 67 - MODEL IDENTIFICATION SCREEN 89
FIGURE 68 - MODEL IDENTIFICATION EXAMPLE 90
FIGURE 69 - MODEL IDENTIFICATION EXAMPLE 2 91
FIGURE 70 - MODEL IDENTIFICATION SLICE SELECTION 92
FIGURE 71 - MODEL STRUCTURE FORM 93
FIGURE 72 - SELECTING THE TRANSFER FUNCTION 94
FIGURE 73 - MODEL IDENTIFICATION FOR INTEGRATING LOOPS 95
FIGURE 74 - GRAPHICAL ASSESSMENT EXAMPLE 97
FIGURE 75 - ZOOMED IMAGE OF THE MODEL PLOTS 97
FIGURE 76 - MAIN EVALUATION SCREEN – TIME DOMAIN 98
FIGURE 77 - MAIN EVALUATION SCREEN – FREQUENCY DOMAIN 99
FIGURE 78 - ADD TUNING 102
FIGURE 79 - ADD TUNING DIALOG 102
FIGURE 80 - TUNING OPERATION 103
FIGURE 81 - MAIN TUNING SCREEN 103
FIGURE 82 - MAIN TUNING SCREEN FOR OUTER CASCADE LOOP 104
FIGURE 83 - TUNING METHOD SELECTION 105
FIGURE 84 - OPTIMAL TUNING PARAMETERS CALCULATION 106
FIGURE 85 – TUNING WITH “TUNING MODEL” AND “SIMULATION MODEL” 107
FIGURE 86 - PERFORMANCE AND ROBUSTNESS PARAMETERS 109
FIGURE 87 – FRAME FOR LAMBDA TUNING 111
FIGURE 88 – FRAME FOR DOMINANT POLE PLACEMENT TUNING 111
FIGURE 89 – FRAME FOR INTERNAL MODEL CONTROL TUNING 112
FIGURE 90 – FRAME FOR LAMBDA-SMITH TUNING 113
FIGURE 91: ITAE TUNING WINDOW 114
FIGURE 92: CONTROL PARAMETERS CALCULATED BY ITAE 115
FIGURE 93: ITAE DESIGN WITH LIMITED CO GRADIENT 116
FIGURE 94: ITAE OPTIMIZATION COURSE 117
FIGURE 95 - FB EVALUATION TAB 118
FIGURE 96 - FF TUNE & SIMULATE TAB 119
FIGURE 97 - FF EVALUATION TAB 121
FIGURE 98 - SAVE TUNING CONSTANTS 121
FIGURE 99 - VIEW LOG FILE 122
FIGURE 100 – TUNING LOG 123
FIGURE 101 - BLOCK DIAGRAM OF CASCADE CONTROL SCHEME 124
FIGURE 102 - CONFIGURATION FORM FOR CASCADE MASTER / OUTER LOOP 125

ABB. All rights reserved

FIGURE 103 - MODELING FORM....126
FIGURE 104 - TUNING FORM FOR CASCADE MASTER / OUTER LOOP....127
FIGURE 105 - DATA FOR A PROCESS....129
FIGURE 106 - LAUNCHING LPM SERVER SERVICE (1)....131
FIGURE 107 - LAUNCHING LPM SERVER SERVICE (2)....131
FIGURE 108 - AUDITING TREE STRUCTURE....132
FIGURE 109 - AUDITING CONFIGURATION....133
FIGURE 110 - AUDITING SHIFT DATA DIALOG....133
FIGURE 111 - AREA CONFIGURATION WINDOW....134
FIGURE 112 - ADD LOOP DIALOG....135
FIGURE 113 - ADD LOOP NAME....135
FIGURE 114 - CONFIGURE LOOP....135
FIGURE 115 - LOOP CONFIGURATION DIALOG: BASE TAGS....136
FIGURE 116 – LOOP CONFIGURATION FOR CASCADE MASTER....137
FIGURE 117 - LOOP CONFIGURATION: AUXILIARY TAGS....137
FIGURE 118- LOOP CONFIGURATION: CONTROLLER CONSTANTS....138
FIGURE 119 - OPC BROWSER DIALOG....138
FIGURE 120 - LOOP CATEGORY CONFIGURATION....139
FIGURE 121 - LOOP CATEGORY CONFIGURATION WINDOW....140
FIGURE 122 - LOOP CATEGORY CONFIGURATION, THE LOOPS TAB....141
FIGURE 123 – BATCH CATEGORY SHIFT CONFIGURATION MENU....142
FIGURE 124 - BATCH CATEGORY/SHIFTS RESUME....142
FIGURE 125 – BATCH CATEGORY SHIFT SETUP....143
FIGURE 126 _PLOT INDICES SELECTION....147
FIGURE 127 - PLOT INDICES: APPEND MONTH MODE....148
FIGURE 128 - STACKED PLOT OPTION....148
FIGURE 129 –COMPARE MONTHS....149
FIGURE 130 - OPTIONS....150
FIGURE 131 - ADD REPORT DIALOG....151
FIGURE 132 - REPORT CONFIGURATION: LOOP CATEGORIES....152
FIGURE 133 - REPORT CONFIGURATION: PLANT AREAS....152
FIGURE 134 - REPORT CONFIGURATION: INDICES....153
FIGURE 135 - REPORT CONFIGURATION: DIAGNOSES....153
FIGURE 136 - REPORT CONFIGURATION: DETAILED LOOP REPORT....154
FIGURE 137 - AREA CONFIGURATION WITH AREA CLASSIFICATION TAG....155
FIGURE 138 - SELECTING CONFIGURE REPORT....156
FIGURE 139 - SPECIFYING DESTINATION FOLDER FOR AUDITING REPORTS....156
FIGURE 140 - EXAMPLE OF WEEKLY REPORT....157
FIGURE 141 - EXAMPLE OF MONTHLY REPORT....157
FIGURE 142 - EXAMPLE OF PERFORMANCE OVERVIEW REPORT....158
FIGURE 143 – EXAMPLE OF THE "*WHAT HAS CHANGED*" REPORT....159
FIGURE 144 - EXAMPLE OF DETAILED LOOP REPORT....160
FIGURE 145 – INTERACTIVE REPORT SELECTION FORM....162
FIGURE 146 – AGGREGATE REPORT, AREA OVERVIEW....163
FIGURE 147 – PREDICTION HORIZON SETTINGS....164
FIGURE 148 – THRESHOLD MATRIX SETTINGS....165
FIGURE 149 – PERFORMANCE WEIGHTS TAB....166
FIGURE 150 – RANK LEVELS TAB....167
FIGURE 151 – MONITOR SERVER STATUS....167
FIGURE 152 – STATUS MONITOR: LPM AUDITING COLLECTIONS LIST AND SCHEDULE....168
FIGURE 153 - STATUS MONITOR: STATISTICS....169
FIGURE 154 - STATUS MONITOR: TAG VALUES....170

ABB. All rights reserved

FIGURE 155 – WEB CLIENT ADDRESS 171
FIGURE 156 – LPM WEB CLIENT LOGIN 171
FIGURE 157 – LPM WEB HOMEPAGE 172
FIGURE 158 - REMOTELY ACCESSING LOG FILES....173
FIGURE 159 - REMOTELY ACCESSING HISTORICAL REPORTS 173
FIGURE 160 - REMOTE REPORT CONFIGURATION....174
FIGURE 161 – CREATING/MODIFYING REPORT CONFIGURATION....175
FIGURE 162 - LPM WEB WITH ADMINISTRATIVE RIGHTS 176
FIGURE 163 – LPM WEB: ADDING USERS 176
FIGURE 164 – LPM WEB: CONFIGURED USERS 177
FIGURE 165 – LPM WEB: MODIFY USER....178
FIGURE 166 – ADD/REMOVE WINDOWS COMPONENTS 179
FIGURE 167 –IIS SELECTION 180
FIGURE 168 – WEB SHARING 181
FIGURE 169 – WEB FOLDER CONFIGURATION....182
FIGURE 170 – WEBCLIENT PROPERTIES – VIRTUAL DIRECTORY TAB....183
FIGURE 171 – APPLICATION CONFIGURATION – OPTIONS TAB 184
FIGURE 172 – WEBCLIENT PROPERTIES – SECURITY TAB....185
FIGURE 173 – SELECT USERS OR GROUPS 186
FIGURE 174 – WEBCLIENT PROPERTIES – SECURITY TAB....187
FIGURE 175. CONNECTING KM 6.1 SERVER TO LPM 3.3....189
FIGURE 176. TESTING KM 6.1 CONNECTION....189
FIGURE 177. CONFIGURING AREA WITH KM 6.1 AS DATA SOURCE....190
FIGURE 178. LOOP CONFIGURATION WITH KM 6.1 AS DATA SOURCE....191
FIGURE 179. DATA COLLECTION FOR TUNING WITH KM 6.1 AS DATA SOURCE....192
FIGURE 180. DATA COLLECTION OPTION DIALOG....192
FIGURE 181. DATA COLLECTION – DATASET DESCRIPTION....193
FIGURE 182. DATA COLLECTION - COMPLETE 193
FIGURE 183. DATA COLLECTED FROM KM 6.1 LOGS 194
FIGURE 184. RAW AUDIT DATA COLLECTION....195
FIGURE 185 FINAL STAGE OF AUDIT DATA COLLECTION FROM KM.....196
FIGURE 186. KPIS AND DIAGNOSTIC IN KM 6.1 DATABASE 197
FIGURE 187. KM AUDIT REPORTS 197
FIGURE 188. REPORT OVERVIEW 198
FIGURE 189. INDEXES IN KM 6.1 REPORTS 198
FIGURE 190. LOOP AUDIT DATA....199
FIGURE 191. LOOP DATA AVERAGED 200
FIGURE 192. DIAGNOSTIC IN KM 6.1....200
FIGURE 193. WORST / BEST LOOPS REPORT....201
FIGURE 194 – LOOP FIND....202
FIGURE 195 – GENERAL TAB 203
FIGURE 196 – AREA ALLOCATION TAB 204
FIGURE 197 – CATEGORY ALLOCATION TAB 204
FIGURE 198 – CONTROLLER ALLOCATION TAB....205
FIGURE 199 – OPTION FORM, TAB 1....205
FIGURE 200 – OPTION FORM, TAB 2....207
FIGURE 201 - ASSIGN TREND TAGLIST 209
FIGURE 202 - CONFIGURE TREND TAGLIST....209
FIGURE 203 - MODEL STRUCTURE....211
FIGURE 204 - FIRST ORDER LAG....212
FIGURE 205 - SECOND ORDER LAG....212
FIGURE 206 - THIRD ORDER LAG 213

ABB. All rights reserved

FIGURE 207 - LEAD LAG....213
FIGURE 208 – LAPLACE TRANSFORM REPRESENTATION....214
FIGURE 209 – LAPLACE TRANSFER FUNCTION FOR ANALYZING FEEDBACK CONTROL....214
FIGURE 210 - FIRST ORDER LAG WITH DEADTIME....215
FIGURE 211 - INVERSE RESPONSE LEAD LAG....215
FIGURE 212 - EFFECT OF Z AND Ω VALUES....216
FIGURE 213 - TAGLIST CREATOR AFTER DATABASE SEARCH....295
FIGURE 214 - MANUAL SELECTION OF CFG FILES....295
FIGURE 215 - LOOPS DETAIL TABLE....296
FIGURE 216 - AREA AND CATEGORY RULES....297

LIST OF TABLES

TABLE 1 – LIST OF COMMON TERMS AND ABBREVIATIONS....14
TABLE 2 - *MINIMUM RECOMMENDED HARDWARE AND SOFTWARE REQUIREMENTS*....17
TABLE 3 – CPMPLUS LPM LICENSES....55
TABLE 4 – LPM CLIENT TOOL MENUS AND MENU ITEMS....61
TABLE 5 - LIST OF AUDITING PERFORMANCE INDICES....144
TABLE 6 - LPM DIAGNOSTIC HYPOTHESIS AND PRE-CONDITIONS....146
TABLE 7 – INTERACTIVE REPORT PROPERTIES....162
TABLE 8 - AREA AGGREGATE REPORT ITEMS....163
TABLE 9 – PRECONDITIONS LIST....165
TABLE 10 – RANK LIST....166
TABLE 11 – GENERAL INFORMATION ON CONFIGURATION....203
TABLE 12 – OPTIONS, FIRST TAB....206
TABLE 13 - DATASET TRENDS RIGHT CLICK MENU ITEMS....208
TABLE 14 – LIST OF INDICES....221
TABLE 15 – SUMMARY OF INDICES....274
TABLE 16 – REGISTRY ENTRIES FOR LPM....299

ABB. All rights reserved

1. Overview

1.1 *About This Manual*

This manual covers the information you need to assess and improve the performance of your control loops through use of cpmPlus Loop Performance Manager (indicated in the rest of this manual with, *LPM*).

A copy of this manual is also provided in the form of an Adobe® Acrobat® PDF file, located in the `Doc` subdirectory of LPM main directory; LPM main directory is selected during the installation procedure.

1.2 *Who Should Read This Manual*

This Manual is intended for those interested in using an efficient tool to create reliable and optimized tuning for basic control loops, and/or to monitor and maintain the performance of control systems at their peaks. This includes control room personnel, process engineers, maintenance engineers, automation and plant managers.

1.3 *Prerequisites*

In addition to a basic understanding of process control, this Manual assumes that you are familiar with:

- The standard Windows user interfaces.
- The format of files generated by plant process data collection and logging facilities.

Before you start using LPM software, please refer to `LPM ReadMe.txt` in the software distribution kit for any additions or corrections to the information in this Manual.

ABB. All rights reserved

1.4 *Document Conventions*

The following type conventions are used to identify special terms or the names of graphical program elements:

1.4.1 Graphical User Interface Elements

- *Italic* type identifies the names of graphical elements such as a section in a dialog (e.g., the *Variables* section).
- The names of dialog boxes are ***bold italic***
- Labels on boxes and names of menus in the menu bar are identified by bold font (e.g., the **Specify Data** box).

`New Courier` font identifies choices in a menu or drop-down list, commands that you enter from the keyboard, or responses such as file names that the computer displays.

1.4.2 Note or Warning

The text **Note** precedes notes and warnings. It identifies a situation that requires special attention, or provides additional explanatory information.

1.5 *Terminology*

The following is a list of terms and abbreviations associated with Loop Performance Manager that you should be familiar with.

Term	Description
CO	Control Output
DCS	Distributed control system
FB	Feedback
FF	Feed forward
IMC	Internal Model Control
KM	Knowledge Manager (to be superseded by cpmPlus History)
Lambda	Common variable name for desired closed loop time constant
LPM	Loop Performance Manager
PDA	Plant-wide Disturbance Analysis
PID	Standard 3 term controller available in all common distributed control systems
PLC	Programmable Logic Controller
PV	Process Value
SP	Setpoint

Table 1 – List of common terms and abbreviations

ABB. All rights reserved

1.6 *How This Manual Is Organized*

This LPM User's Manual contains these major sections:

- Chapter 1, *Overview*, contains an overview of general concepts
- Chapter 2, *Introduction*, contains an overview of LPM functionalities.
- Chapter 3, *Installing LPM,* shows how to install LPM components.
- Chapter 4, *Configuration/Application Building* contains a description of the LPM Interface.
- Chapter 5, *Tuning*, provides information on how to perform tuning with LPM
- Chapter 6, *Auditing*, provides information on how to perform Auditing with LPM.

- *Appendix A* contains details to operate with charts.
- *Appendix B* contains details on Laplace Transform.
- *Appendix C* Contains technical details about tuning evaluation parameters.
- *Appendix D* contains a detailed description of the performance indices calculated by LPM
- *Appendix E* contains a detailed description of the diagnosis calculated by LPM
- *Appendix F* contains a detailed description of the PID controllers types supported by LPM
- *Appendix G* contains a detailed description of the Taglist Creator module for Infi90 and AC800F
- *Appendix H* contains a detailed description of Windows registry keys that can be used to customize LPM

A separate manual describes *Plantwide Disturbance Analysis* (PDA) features.

ABB. All rights reserved

2. Introduction

2.1 *Product Overview*

cpmPlus^{IT} Loop Performance Manager is a tool for optimization of process control performance. Its users are control engineers, instrument technicians and maintenance personnel, whose task is to ensure that control loops operate at peak performance. The package consists of two online software applications: an auditing module used to monitor loop performance and a tuning module used to improve control loop performance. Any industrial plant can benefit from Loop Performance Manager, and LPM achieves these benefits without requiring a large capital investment.

Improved PID controller performance can result in reduced quality variability, reduced energy consumption and reduced downtime. The ability to maintain PID loops at their optimal operating conditions translates into manufacturing cost savings and higher productivity. Optimizing control loop performance, however, can be a difficult and a time consuming task, even for an experienced control systems engineer or instrument technician. Many plants forgo potential economic benefits from their distributed control system (DCS) investments because of poorly tuned or manually operated loops.

cpmPlus Loop Performance Manager provides powerful, yet extremely easy-to-use, software tools to diagnose and improve control loop performance. The package supports tuning for a wide variety of commercial PID controllers (both from ABB and others), as well as various configuration options for the PID control algorithms. By combining data collection, modeling, tuning and simulation in a single user interface, ABB delivers a complete tool and removes the barriers preventing optimization of DCS investments.

The auditing tool runs in the background periodically collecting data for multiple loops, analyzing the data and creating reports for plant personnel to review. The reports help focus plant maintenance efforts by highlighting the control loops with the poorest performance and diagnosing the problems. This information allows the plant personnel to investigate and repair equipment problems, retune control loops and improve their performance.

The tuning tool supports online data collection through OPC. The data can be collected while the PID loop is in either MANUAL or AUTO mode, i.e. the user may make changes to the setpoint in AUTO mode or to the controller output in MANUAL mode. The data is used to build a model and the model is then used to tune the control loop. The user selects a tuning method and specifies desired performance parameters, and then the tool calculates the optimal tuning constants for the selected industrial PID controller.

cpmPlus Loop Performance Manager is intended to run on a personal computer (PC) under Microsoft® Windows® Operating System. The online applications communicate with the OCS/DCS through an OPC Data Access (DA) client. The OPC client can communicate with multiple OPC servers, which may be on the same PC or located across the network and are compliant with OPC DA 1.0 or 2.0 specifications. However, in case the OPC server is not compliant with OPC 1.0 specification, it may not be possible to perform automatic OPC server search. In this case the connectivity parameters (OPC name, PC network address) must be manually entered.

ABB. All rights reserved

2.2 *Prerequisites and requirements*

2.2.1 *Workstation Hardware*

Table 2 - *Minimum Recommended Hardware and Software Requirements* lists the minimum recommended hardware and software requirements for Loop Performance Manager.

Item	Requirement
Primary memory	1GB minimum
Operating system	Microsoft Windows XP with Service Pack 2 or Windows 2003
Other software required	• Microsoft XML Version 3.0 Service Pack 2 or later[2] • Microsoft Data Access Components (MDAC) Version 2.6 or later [3] • OPC DA server (required for data collection) • Microsoft Excel 2000 or 2003 (required for loop auditing reports) • Microsoft Dot Net Framework 2.0 • KM-LPM Extension (For KM 6.1 support (See Chapter 7.0)

Table 2 - *Minimum Recommended Hardware and Software Requirements*

NOTES:

1. The selected personal computer must be designed to withstand climatic, mechanical, and electrical effects in the intended environment.
2. If Microsoft XML software is not available on target PC the setup program will install it.
3. If Microsoft MDAC software is not available on target PC the setup program will install it.

2.2.2 *Connectivity – OPC case*

It is assumed availability of an OPC Server with OPC Data Access 2.0 standard.

2.2.3 *Connectivity – Infi90 case*

It is assumed availability of one of the following Infi 90 cards

- ICT03A/13A
- ICT12

Use of SCSI connection is strongly suggested. The use of a serial link may reduce the available bandwidth. It is assumed availability of semAPI software and license.

ABB. All rights reserved

2.2.4 Connectivity – KM 6.1 case

It is assumed that a Knowledge Manager (KM) server with version 6.1 is installed on a separate node. Also, it is required to log the plant data signals in the KM database for using the KM database as data source for auditing or tuning purposes. See section 7 for more details.

2.3 Intended user

This manual is intended for users responsible for installation, configuration, runtime operation or maintenance of Loop Performance Manager. To tune control loops and understand auditing reports, the user should have basic knowledge of DCS control systems, feedback control loops and field instrumentation. Basic Windows skills are also required.

This manual is not intended to be the sole source of instruction for this product. Users involved in controller tuning and maintenance should attend the applicable course offered by ABB

2.4 Product description

cpmPlus Loop Performance Manager (LPM) consists of two software applications. The first is a loop tuning package that runs on demand when the user wants to determine the best tuning parameters for a PID control loop. The second portion is a performance auditor that runs in the background, monitoring the behavior of the control system across a process or a large section of an operating plant and producing assessments and diagnostics on the status of the plant loops. The auditing function periodically creates reports that plant personnel should review and act on to maintain peak process performance.

2.4.1 Loop Auditing

The Loop Auditing component executes the following procedures:

- Periodic data collection for monitored loops
- Performance index calculations
- Archiving and plotting of performance indices
- Archiving and plotting of diagnosis
- Report generation

To setup auditing, the user organizes control loops into common categories that determine the sampling frequency and duration of the periodic data collections. Loop auditing begins with daily periodic data collections that operate in the background for up to 1000 loops per LPM server, with the maximum value depending on the frequency of data collection, see Table …. After data collection is complete, performance indices are computed and stored in the auditing database. The performance indices may be plotted over time for multiple loops to view historical changes. The indices are explained in Section 12.

To create a plot, use the plotting icon on the tool bar to bring up the screen shown in Figure 1. Select the tags to plot from the list on the left and the performance indices from the list in the center. On the right end a list with all the months for which Performance Indices are available is shown.

ABB. All rights reserved

Note that Months can be appended (as in Figure 1 - Audit Plots) or compared, depending by which radio button has been selected.

By using the Ctrl Key or the Shift Key it is possible to select different or multiple months.

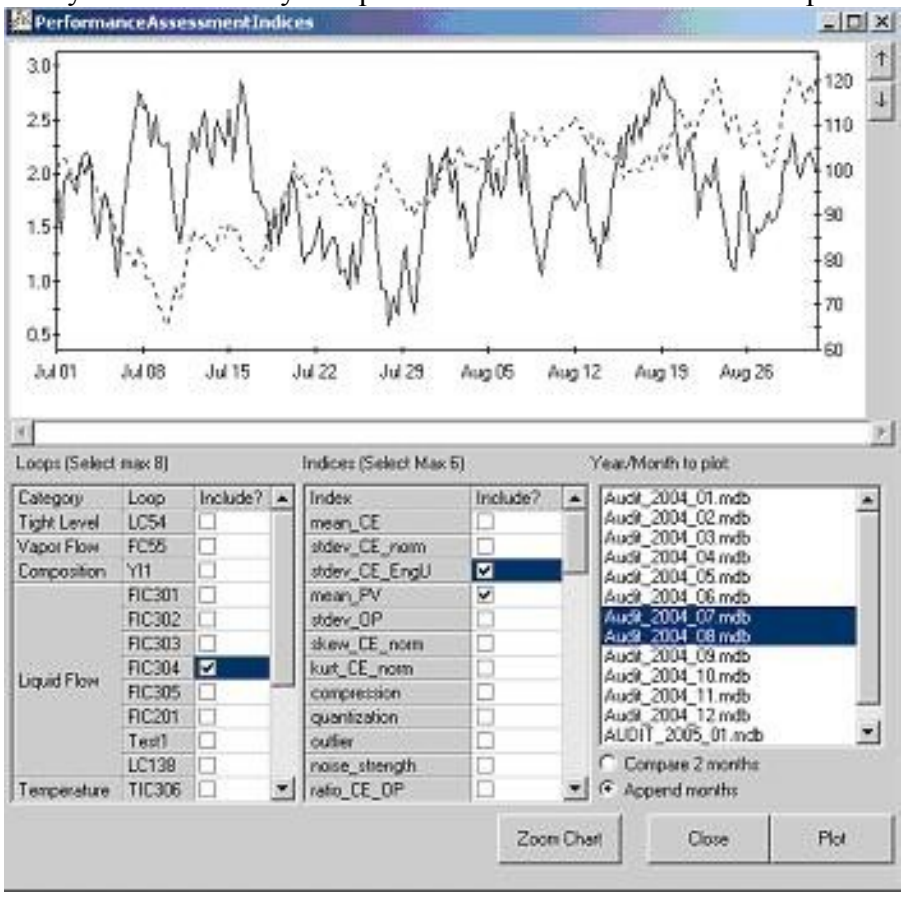


Figure 1 - Audit Plots

A similar plotting facility is available for Diagnosis.

The audit reporting function automatically generates a loop ranking report once per week as an Excel spreadsheet file. This report graphically indicates the percentages of loops with performance classified as: Excellent, Good, Fair, and Poor. This report also highlights each loop identified as performing poorly by indicating the likelihood of specific problems, such as sluggish tuning, valve stiction, loop oscillation or other causes. The problem diagnoses are determined from the evaluation of rules involving the performance indices.

The user may configure additional reports by choosing which plant areas, loop categories, performance indices and diagnoses to display, as shown in Figure 2. LPM automatically generates these reports as Excel spreadsheet files on a weekly basis. The user may also create on-demand reports. The reports provide historical performance records and enable the maintenance team to focus on the most important process control problems.

ABB. All rights reserved

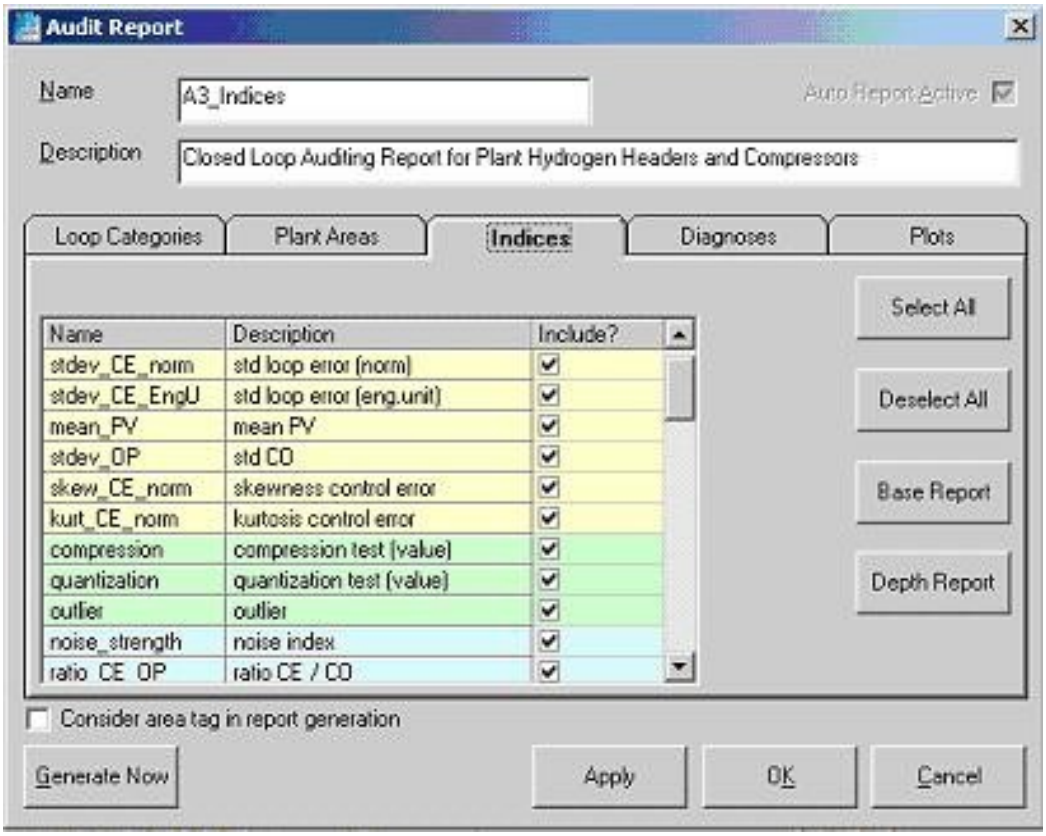


Figure 2 - Report Configuration

LPM makes available eleven diagnoses computed from the forty-six performance indices.

The last tab (Plots) allows specifying up to 10 loops for which a detailed report is requested. Detailed Report includes a number of plots and trends in order to perform in-depth, sophisticated analysis both in the time and in the frequency domain.

2.4.1.1 Auditing Status Monitor

LPM is provided with a facility to monitor the performance of the various auditing functions. This facility can be launched by the user via the Tools menu option “Monitor Server Status”.

2.4.2 Control Loop Tuning

The effectiveness of a process operation depends greatly on the automation system in place. The automation system may be a DCS, single loop controllers or a PLC. Regardless of what type of system is used, the plant personnel must tune the controllers to keep the process operating at peak efficiency. Over a period of time, the process may show signs of physical wear or the normal operating point of the plant may shift. These sorts of changes degrade the performance of the control loops, requiring them to be returned.

With LPM, the maintenance personnel can easily tune PID loops to regain lost performance. The procedure is simple:

- Configuration: configure the control loop tags and type.

ABB. All rights reserved

- Data Collection: collect data showing how the measured values react to actuator steps.
- Modeling: with the modeling tool create a linear model that “explains” the collected data reasonably well.
- Select a tuning rule, specify desired performance parameters and then calculate new PID tuning parameters

LPM is launched from the start menu or from a desktop icon. The user then navigates the software using an object tree. The tree is organized by plant areas. The user adds control loops to the desired area. The configuration requires the tag names and parameter extensions for setpoint, process value, and control output for each loop.

2.4.2.1 Data collection

Once a loop is configured, the user may proceed to the Data branch of the tree.
Clicking on the Data item allows the user to review an old data collection or to launch a new one.
When the user starts a new data collection, LPM displays the data on a trend plot. The user may launch several data collections for different loops simultaneously. Each data collection is stopped automatically when the specified time duration is reached. Alternatively, a data collection can be stopped manually by the user. When complete, the data can be saved for modeling or for plotting at a later time.

ABB. All rights reserved

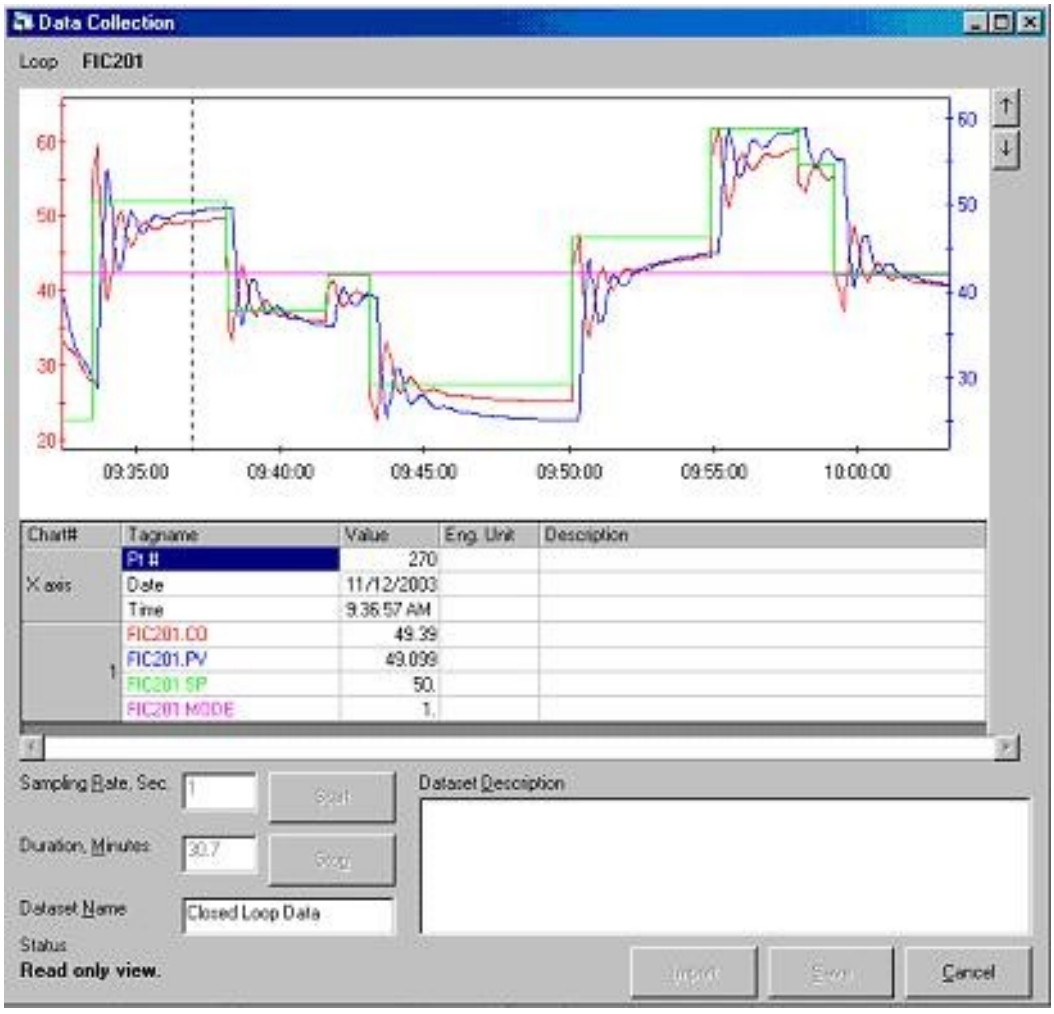


Figure 3 - Data collection

The user must ensure that there is sufficient excitation (information) in the collected data to successfully build a model with the model identification tool. The user must either put the loop in Manual mode and step the value of the control output, or change the setpoint of the loop while in Auto mode.

Typically there should be a minimum of four step changes in the data if the automatic model building option is used for model identification. If the process is noisy so that the effects of the steps are not apparent, or if the steps do not provide consistent results, then more data should be collected.

2.4.2.2 Model identification

This section refers to Laplace transform models. Appendix B provides a short review.

After data collection, the user may now launch the modeling tool from the navigation tree. The Modeling form comes in two different settings:

- the **Basic** one, for the occasional user, provides a quick and more intuitive way to build models (Figure 4);

ABB. All rights reserved

- The **Advanced** one, for the most experienced users, which makes available several knobs and options, for more control of the model building procedure Figure 5).

The user can easily toggle from one setting to the other just clicking on the toggle button.

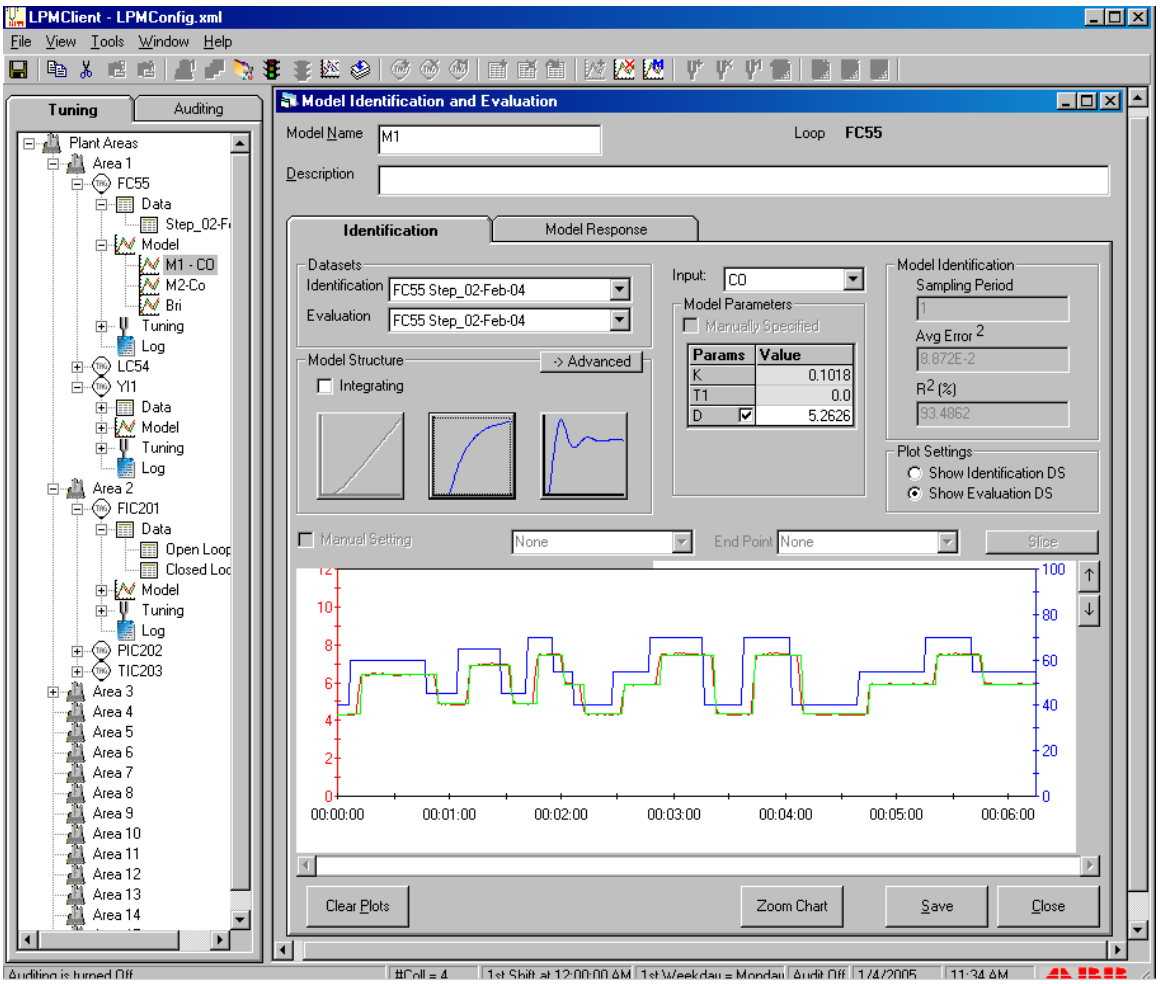


Figure 4 - Model identification: the Basic form

ABB. All rights reserved

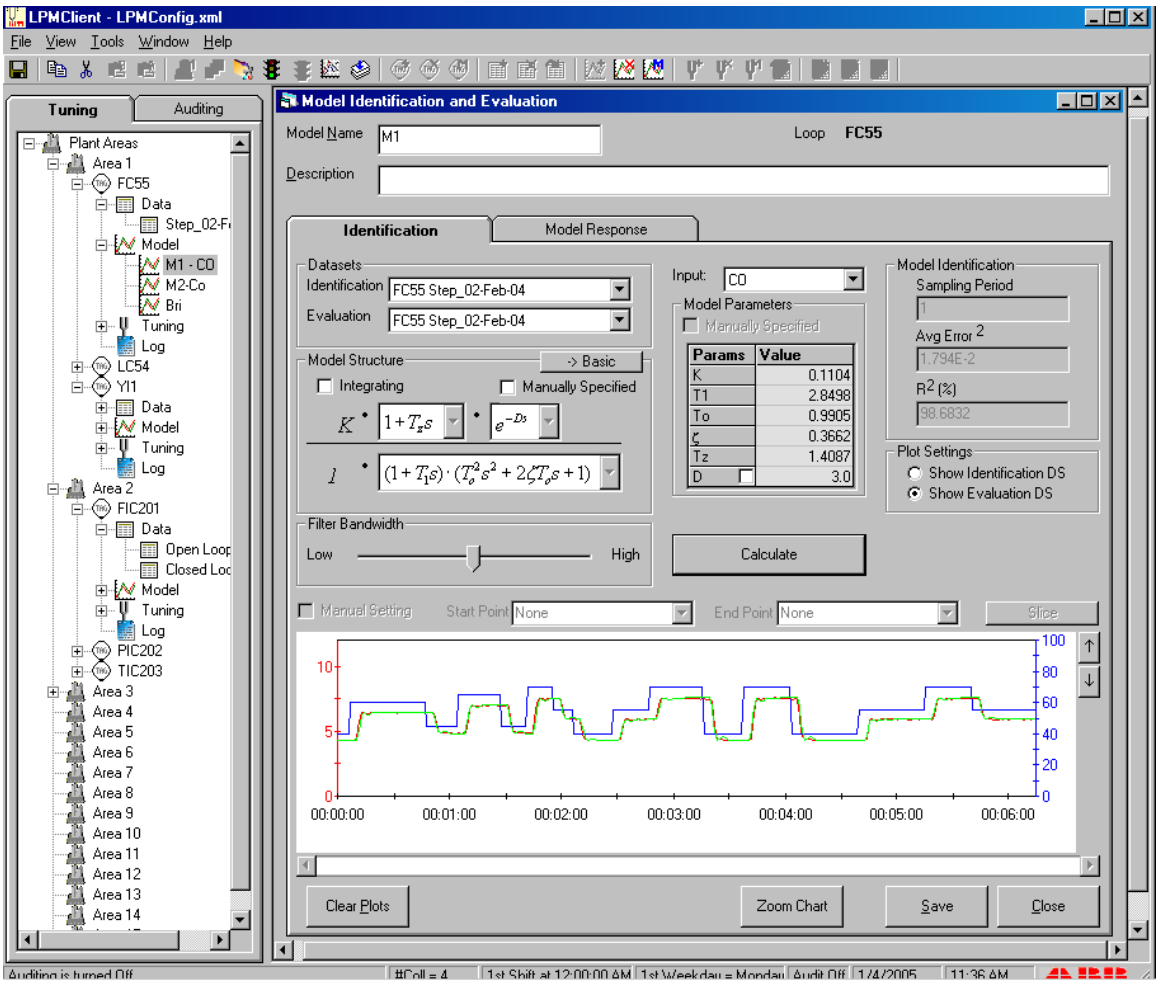


Figure 5 - Model identification: the Advanced form

Notwithstanding which form is selected, the user is requested to specify two data sets, one for model identification and the second for model evaluation. The evaluation set is used for the calculation of the model statistics in the right panel. The plot area may be set to display the identification data set or the evaluation data set by using the radio button above the plot. When the evaluation set is plotted, the model predictions (i.e. process values calculated by the model) are also displayed with the raw data.
If the user selects the same data set in both drop down boxes, the software automatically splits the data set, using the first half for identification and the second half for assessing the models. The identification algorithm then re-builds the model using the best structure and all the data.

Common to the forms is also the possibility to remove part of the data from the identification procedure. This is a very useful feature when some data are clearly corrupted and must not be used for model building purposes. When showing the identification dataset, the algorithm allows removing data at the beginning and/or at the end of the trend in either a graphical (acting on the "`Slice`" button) or an analytical way (checking the "`Manual setting`" checkbox and selecting start and end point). A colored area will highlight the data kept for identification (Figure 6).

ABB. All rights reserved

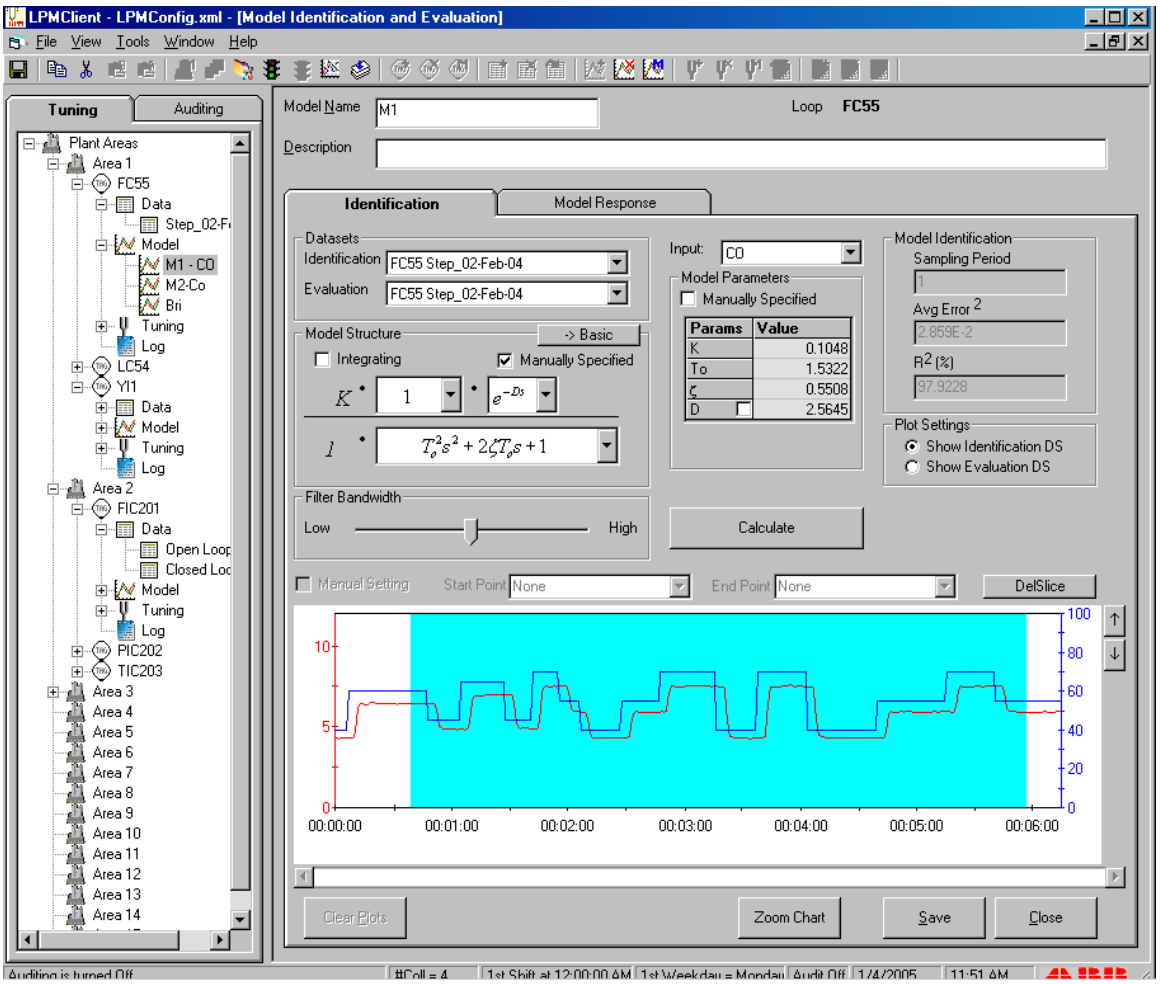


Figure 6 - Removing Bad Data from the identification dataset

Building models in the Basic form is really straightforward. After selecting the dataset(s) the user has just to specify is the process is integrating (i.e. level control) or not acting on the related checkbox. Depending on this check the first one or the last two icons will be highlighted. If the process is an Integrating one, just clicking on the first icon will produce the model. If it is not the user can choice between the simplest shape (correspondent to a first-order model with pure delay), or the more complex one, related to a second-order model. The only additional knob available in this form is the possibility for the user to explicitly provide the delay time, acting on the checkbox, in the "`Model Parameters`" area.

A more thorough approach to Model Identification is provided in the "Advanced" form.

When the manual check box is marked in the Model Structure section, the user selects from several drop down lists to create a general Laplace transform model (see Appendix B). The first box in the numerator allows the selection of a lead term (Tz s+1) or 1. The second factor allows the selection of a deadtime or 1. The first factor in the denominator can be set to s, indicating an integrator or a 1. The last box gives several choices for the denominator as shown on the right.

ABB. All rights reserved

Note that the complete model structure supports simple first order models, up to a third order model with integrating and deadtime options. The Tz parameter in the numerator can be negative, allowing the algorithm to fit an inverse response model. The denominator can include a general quadratic form, allowing for underdamped models.

When clicking the Calculate button with a manually defined model structure, the identification algorithm finds the best values for the model parameters, (gain, deadtime, and time constants) that best explain the data. There is also an option to select these model parameters manually.
Note that in the Advanced view, the evaluation data set has one more purpose. If the user lets the software select the model structure, then it uses the evaluation data set to calculate model fit statistics and select the best model.

The user may experiment with the different modeling options and plot the results in the trend area for comparison. The model fit statistics also provide indications of the model accuracy relative to the evaluation data. The evaluation tab shows a step response plot or frequency response plot for the identified model. When the user is satisfied with the results, the model information can be saved. Each saved model object includes all of the information contained on the modeling form, which consists of dataset names, model structure selections and model parameters. When a model is saved, the model name appears in the object tree below the corresponding loop.

There are a few points to keep in mind when building models. The first task inside the algorithm is estimating the deadtime. If a process includes both deadtime and inverse response, the deadtime estimate will be approximately equal to the real deadtime plus the time for the inverse response to cross the axis. If the resulting model parameters include a Tz numerator term being greater than 0 (no inverse response) when user expects an inverse response or when he clearly sees an inverse response in the data, then the user may want to manually set the deadtime to a smaller value than the initial estimate and re-identify the model. The user may also want to specify the deadtime when identifying a model from closed-loop data. Note that the deadtime parameter can be manually specified even if the algorithm is determining the model structure.

Another point to consider is the effect of noise on automatic selection of the model order. If the data is noisy and the user is letting LPM determine the model order automatically, the resulting model may be of high order (with many parameters). Use the evaluation tab to look at the step response of the model. If the step response indicates that a low order model would work (see Appendix B), then manually select a simpler model structure and identify the model again.

ABB. All rights reserved

2.4.2.3 PID Tuning

With a model, or several models saved, the user then selects the tuning option from the tree.

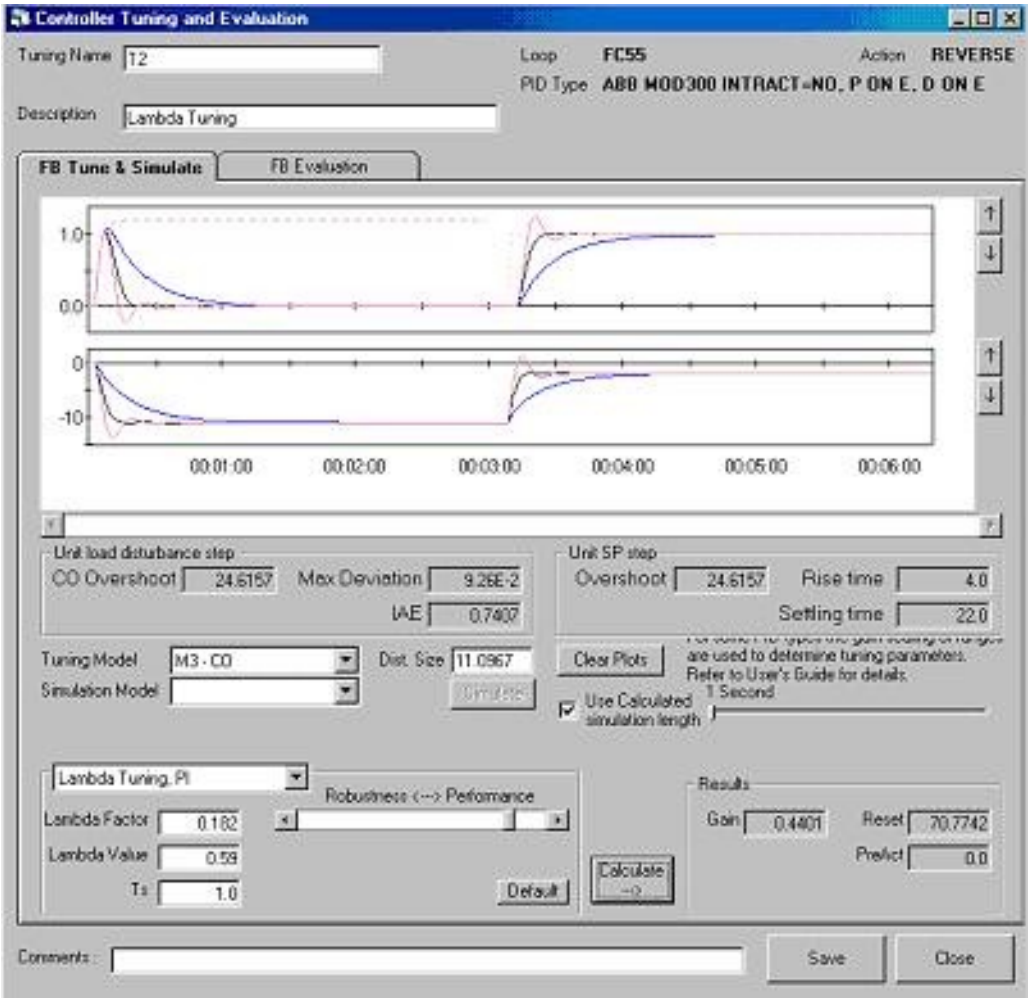


Figure 7 – PID Tuning

To tune a control loop, the user performs the following steps:

- Choose a model
- Choose a tuning method
- Specify the desired tuning performance parameters
- Click the Calculate button

The following tuning methods are available:

- Manual Tuning
- Lambda Tuning PI
- Dominant Pole Placement (DPP) PI
- Dominant Pole Placement (DPP) PID
- Internal Model Control (IMC) Tuning for PI

ABB. All rights reserved

- Internal Model Control (IMC) Tuning for PID
- Lambda-Smith
- Integral Time absolute Error (ITAE)

Note that when selecting the tuning method, the user also specifies whether a PI or PID controller is desired. Each tuning method includes one or more tuning performance parameters to specify the desired levels of performance or robustness for closed-loop control. For each parameter, the user may specify a numerical value or adjust a slider to increase robustness or increase performance.

Manual Tuning

Manual tuning is a useful feature mainly to evaluate the controller performance with the actual tuning parameters against the suggested optimal ones. It could be also used for investigating the effect of manual adjustment on the parameters themselves. If this option is selected LPM allows to copy the existing parameters (inserted in the Loop Configuration activity) into the tuning form so to be able to simulate the actual controller response (Figure 8).

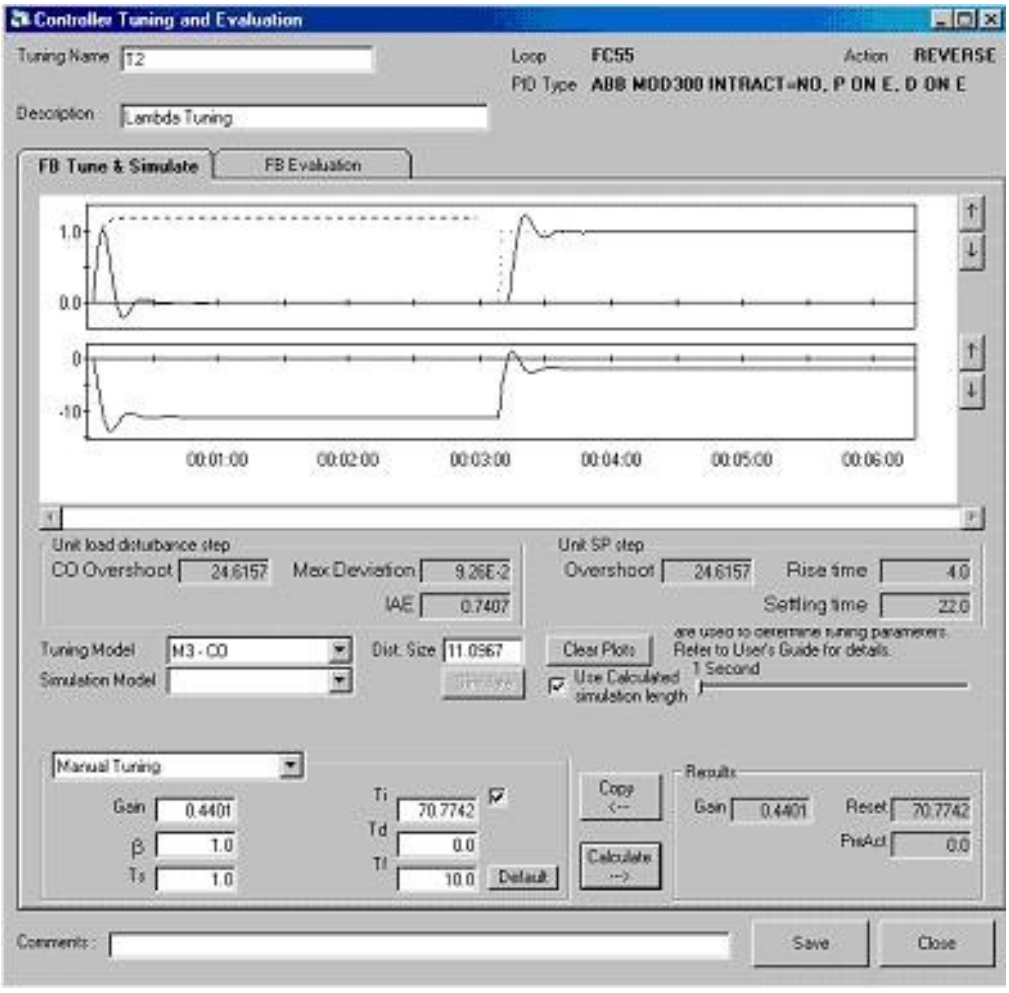


Figure 8 - Manual Tuning in LPM

ABB. All rights reserved

Lambda and IMC tuning
The Lambda tuning method and the IMC method are closely related as the user specifies the desired time constant (lambda) for closed-loop control. A smaller lambda is specified for increased performance (faster control response) or larger value for increased robustness. The Lambda tuning method is restricted to first order plus dead time models and to PI controllers. The IMC tuning supports all model types (provided it is open-loop stable or has integration) and both PI and PID controllers.

Dominant Pole Placement
The dominant pole placement method is based on specifying the desired damping ratio and natural frequency for a second order system representing the dominant dynamics of the closed-loop control (the dominant poles of the transfer function). A smaller damping ratio is specified for increased performance (faster control response) or larger value for increased robustness (less overshoot). A larger natural frequency is specified for increased performance (faster rise time) or smaller value for increased robustness. This tuning method also applies to all model types and both PI and PID controllers.

Integral Time Absolute Error (ITAE)
There are several ways to specify quantitatively loop performance. A common way is to use rise time and overshoot of the process value from a step response. Another different approach is to define a control system as optimum, when the system parameters are adjusted so that a chosen performance index reaches an extreme value, commonly a minimum value. Useful performance indices must be numbers that always are positive or zero. Then the best system is defined as the system that minimizes such an index.
One common performance index is the integral of square of the error (ISE):

$$ISE = \int_0^T e^2(t)\, dt; \quad e(t) : \text{control error}$$

An improved performance index is the integral of time multiplied by absolute magnitude of the error (ITAE):

$$ITAE = \int_0^T t\left|e(t)\right|\, dt; \quad e(t) : \text{control error}$$

The multiplication by the time reduces the contribution of the relatively large initial error value of the ISE, as well as emphasizes errors occurring later in the response. ITAE optimized systems shows a good transient response. They have typically a short well damped overshoot. It was common to calculate the ITAE for small systems analytically to get simple design descriptions for such type of systems. Some tuning methods base on such an approach.

ABB. All rights reserved

The increase of computer power was one reason to extend integral performance indices by additional limitations. Such indices are normally not calculable analytically. Therefore, numerical solutions are needed. The following performance index is a useful extension of ITAE and supported by LPM:

$$\int_0^T t\,|e(t)|\,dt + p\left(\left.\frac{dy}{dt}\right|_{\max}\right); \quad y(t): \text{Control variable}$$

"p" is a constraint and it limits the gradient of the control variable to a user defined maximum. In that way, it is possible to reduce the dynamic purpose fully. That may be useful when the actuator has a limited control time and is not part of the system model.

In Figure 9, very schematic indications about the tuning methods available in LPM are summarized.

Particular care must be used especially in assessing the “Control Task” column: the indications are quite rough and must not be taken as strict rules. Dominant Pole Placement and IMC can provide excellent results for both disturbance rejection and setpoint tracking, but the specific design procedures may sometimes favor the control objective reported in the table.

ABB. All rights reserved

Choice of tuning method - guidelines

	Process model	Control task	Design parameters	Remark
Manual	All	Any – but it is up to the user to know how to tune	Controller parameters	From scratch, or adjustments from the result of automatic tuning
Lambda Tuning, PI	First order, stable (Self-regulating with a single time constant, or purely integrating)	Setpoint tracking	Desired closed-loop time- constant (Lambda or Lambda factor)	Lambda is the desired closed loop time constant Lambda factor relates it to the open loop speed
Dominant pole placement, PI or PID	All	Disturbance rejection (and setpoint tracking, see remark)	Damping ζ of transients and their speed restriction ω_{max}	Universal tuning for both tasks, for controllers with tunable SP weigh
IMC, PI or PID	Stable (Self-regulating or integrating)	Setpoint tracking	MS (Maximum sensitivity) or Lambda	Specifying Ms provides total robustness focus
Lambda – Smith	To be used for Bailey ISC controller only	Setpoint tracking	Desired closed-loop time-constant	To be used on processes with significant delays
ITAE	All	Minimize cost function	Gradient of controller output	Request initial controller parameters that stabilize the loop

For all these tuning methods, the process model is allowed to include a deadtime

© ABB Corporate Research- 1 -

Figure 9 – Tuning Method Guidelines

When clicking the Calculate button, the tuning algorithm applies the selected method and performance parameters to calculate the PID tuning parameters. The resulting PID tuning parameters are displayed using the names and units of the specific PID controller type selected for the loop. The PID type is defined in the loop configuration form. LPM supports more than 40 specific PID controller types.

Attention: ***Most PID controllers also include some form of scaling to normalize the equations and the resulting gain parameters. Many controllers use the PV range and CO range for scaling, while some controllers include a separate gain scaling parameter. In order for LPM to obtain proper tuning parameters for industrial PID types, it must obtain the appropriate PID loop range values or gain scaling parameters. For many PID types, LPM defines the scaling by applying the PV range and CO range defined in the LPM Loop Configuration form. For PID types that use a separate scaling parameter (e.g. Bailey), LPM defines the scaling by applying the gain scaling parameter defined in the LPM Loop Configuration form.***

ABB. All rights reserved

The software also supports three general forms for the PID controller type: ideal, series, and parallel. The various industrial forms of the PID controller have a similar structure to one of these three general forms.

Ideal Form:

$$CO = Gain\left(1 + \frac{1}{T_i s} + \frac{T_d s}{1 + T_f s}\right)E$$

Series Form:

$$CO = Gain\left(1 + \frac{1}{T_i s}\right) \bullet \left(\frac{1 + T_d s}{1 + T_f s}\right)E$$

Parallel Form:

$$CO = \left(Gain + \frac{K_i}{s} + \frac{K_d s}{1 + T_f s}\right)E$$

Where

$E = SP - PV$ denotes the control error

After calculating the tuning constants, the software performs a simulation of the model and the PID controller. The results are displayed in a trend plot. The simulation includes a load disturbance for the first part of the time response and a setpoint change for the second part. Numerical performance indices are also computed and displayed for each part of the simulation (such as overshoot, rise time, settling time, Integrated Absolute Error, etc.).

The tuning sliders can be adjusted to increase robustness or performance and then new tuning values can be recalculated. The trend plots support multiple traces to allow graphical comparisons of the simulation results for the new tuning parameters relative to all the previous results.

To increase tuning reliability LPM makes available the possibility to test loop performance against a model different from the one used to determine the optimal tuning parameters.

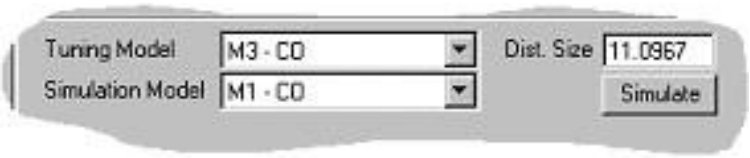


Figure 10 – Test Performances Against a Model

This is possible just selecting a different model in the "`Simulation Model`" drop down list, highlighted in figure 10, and clicking on the "`Simulate`" button.

ABB. All rights reserved

Another improvement in LPM 3.3 is represented by the possibility to change the simulation duration so to properly depict the interesting dynamics. Figure 11 draws attention to the checkbox and the slider, which determines the prediction length and its effect on the simulation trends: the trends at the bottom of the figure have a simulation length around 1/3 of the ones at the top and the magnifying effect is a clear advantage to spot and understand detailed dynamics.

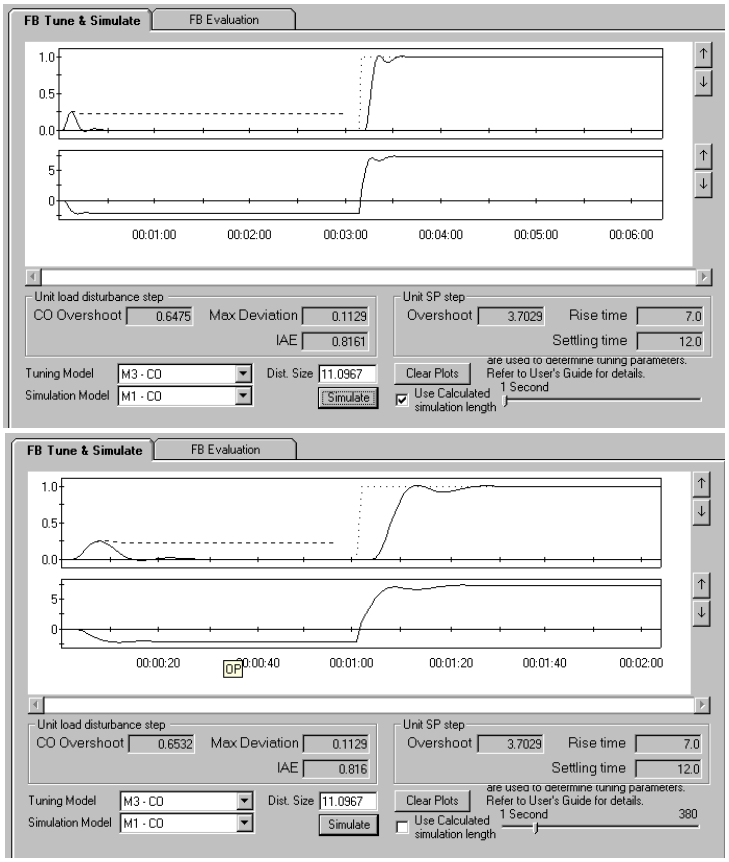


Figure 11 – Change the Simulation Duration

A frequency response plot of the sensitivity function is also available on the Evaluation tab. The sensitivity plot indicates how much a disturbance of each frequency is multiplied by the control system. The maximum value indicates the maximum sensitivity of the control system to a disturbance. A typical design objective for robust control is to get a relatively flat response for higher frequencies and minimize the peak magnitude of the sensitivity function.

All of the information specified in the tuning form can be saved as a tuning object for further retrieval. The user may also store the resulting tuning parameters and related information in a tuning log file that includes a date/time stamp for each change. The log file can be accessed from the object tree.

2.4.2.4 Feedback Tuning rules

The name Lambda tuning for controllers comes from an article written by Dahlin in 1968 [1]. The article discussed tuning digital controllers based on discrete-time transfer functions. The proposed

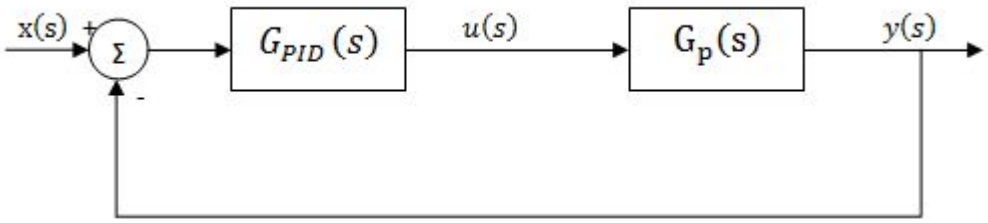
ABB. All rights reserved

algorithm was tuned using a single parameter, λ that was the desired closed loop response time for a first order system. This idea was later expanded to standard PID controllers.

Before discussing lambda and IMC tuning, a little background on IMC is necessary. Figure 13 shows a block diagram for an IMC control loop. An IMC controller is defined using Laplace transfer function models. The controller is made of two parts, Gc, the control algorithm, and Ĝp, the model of the process. Gp represents the real process. IMC controllers are attractive for several reasons:

- They perform well when the process has long deadtimes
- They are tuned by specifying the closed loop response time (λ)
- They can withstand modeling error; a larger λ increases the robustness of the controller.

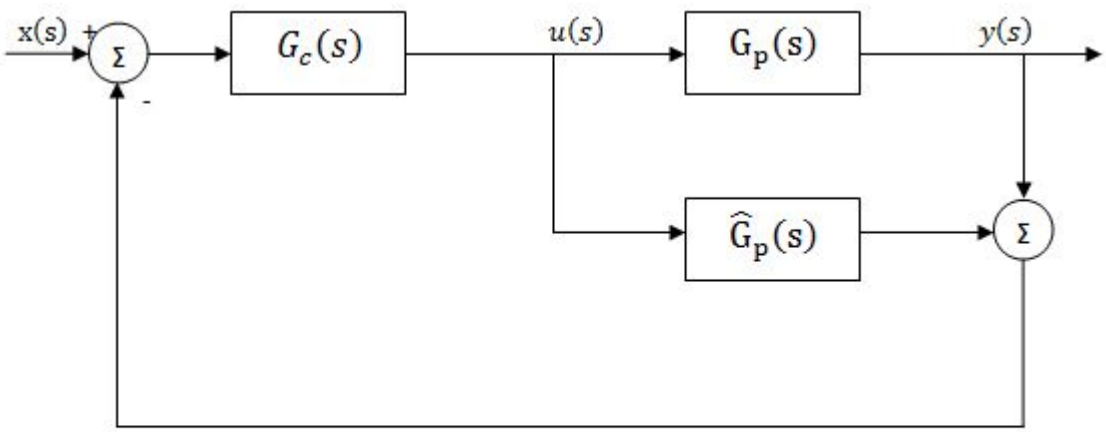
To understand how IMC tuning works, look at the two block diagrams below. Figure 12 is for a PID controller and Figure 13is for an IMC controller.



$$y(s) = \frac{G_{PID}(s).G_p(s)}{1 + G_{PID}(s)G_p(s)}\, x(s)$$

Figure 12 - Block diagram of a PID control loop

ABB. All rights reserved



$$y(s) = \frac{G_c(s).G_p(s)}{1 + G_c(s)\left(G_p(s) - \hat{G}_p(s)\right)} x(s)$$

Figure 13 - Block diagram of an IMC loop

The closed loop transfer function (Laplace transform) mapping the setpoint to the process variable is shown with each diagram. The desired closed loop response is represented as a first order transfer function, F(s), with a time constant of λ. If the equations are set equal to each other and simplified by assuming Gp(s) = Ĝp(s), then using IMC rules to define Gc(s) = F(s)/ Ĝp(s), with the resulting transfer function of the controller is

$$G_{PID}(s) = \frac{F(s)}{\hat{G}_p(s)\left(1 - F(s)\right)}$$

If the process model Ĝp is first order, then the right side of the equation above is a PI controller exactly. The PID controller gain is the ratio of the time constants and process gain, $\tau_p/k_p\lambda$, and the integral time constant is τ_p using the ideal form of the PID equation The Lambda PI tuning method used in LPM supports first order plus deadtime models. The user may specify either the lambda value (desired closed-loop time constant) or the lambda factor, which is the ratio of the closed-loop time constant to the open-loop time constant. The tuning method calculates the PI controller parameters to meet the specification.

The tuning rules labeled IMC PI and IMC PID extend the lambda tuning methods to higher order models. The user may specify either the desired closed loop response time constant (lambda) or the desired peak magnitude of the sensitivity function (Ms), which is shown in the frequency plots (see Evaluation tab). The peak value must be greater than or equal to 1. When Ms is specified, the algorithm iterates on λ to find the tuning constants such that the maximum sensitivity matches the user-specified value.

ABB. All rights reserved

The dominant pole placement algorithm assumes the desired closed loop response is given by a second order transfer function, as seen in the equation to the left. The user specifies the desired closed-loop response by setting values for the damping ratio, ζ, and for the natural frequency, ω. The algorithm determines the tuning constants that approximate the specified values.

Usually, a desirable value for the damping ratio is between 0.6 and 0.8. A damping ratio of 0.7 provides approximately 5% overshoot for a true 2nd order system. Larger values of ζ reduce the overshoot and provide more robustness, but also slow down the response. For ζ > 1, there is no overshoot and the response becomes more sluggish with increasing ζ. The natural frequency also affects the speed of response, since it corresponds to time scaling. The maximum natural frequency in dominant pole design is, however, typically large enough to not be a limiting factor and hence the damping is a more useful design parameter.

A larger ω should be specified to get a faster response, e.g. shorter rise time. However, the effects of ω on the rise time are also dependent on the value of ζ. For example, if □ is increased, then ω should also be increased in order to keep the rise time unchanged. In some cases, large changes in ω are required to increase the speed of response perceptibly.

2.4.2.5 Feedforward Tuning

Loop Performance Manager also provides automatic tuning and analysis for feedforward (FF) control loops.

Figure 14below shows a block diagram of a feedforward control system. LPM calculates an ideal FF compensator ($G_{FF}(s)$), based on the idea to design the transfer function $G_{FF}(s)$ able to cancel the effect of the Disturbance d(s) on the output y(s). This transfer function is given by $G_{FF}(s) = -G_d(s)/G_p(s)$.

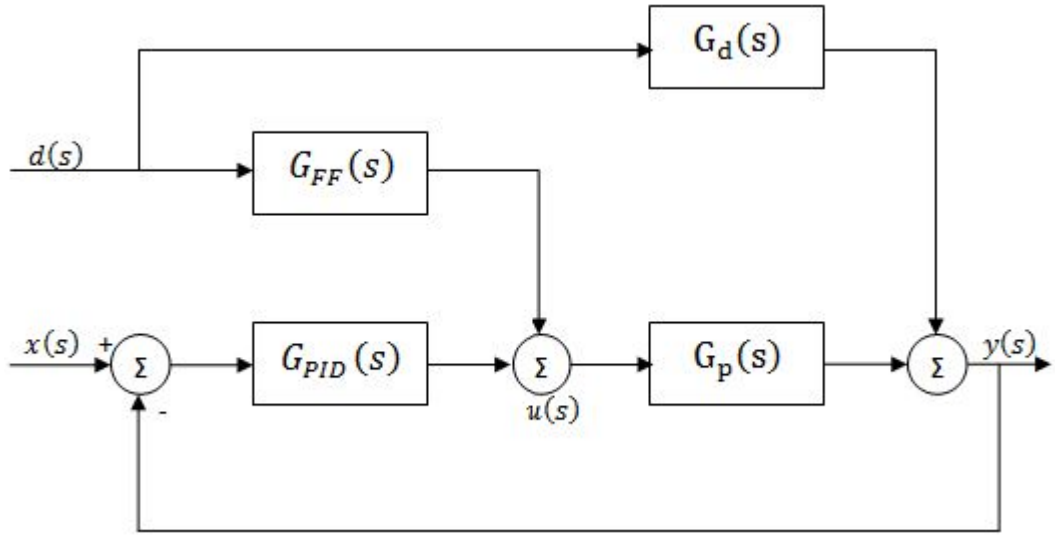


Figure 14 - Feedforward Controller

ABB. All rights reserved

To define a feedforward control loop, the user performs the following steps:

1. Define auxiliary (feedforward) variable(s) in the "loop configuration" form
2. Make a data collection, where the process excitation is generated by the Control Output (CO).
3. Build a model from the control output to the process variable (block $G_p(s)$ in Figure 14) using the data collection of point 2.
4. Make a data collection, where the process excitation is generated by the Disturbance signal ("d" in Figure 14).
5. Build a model from the disturbance signal to the process variable (block $G_d(s)$ in Figure 14) using the data collection of point 4.
6. Tune the feedback portion of the control loop (block $G_{PID}(s)$ in Figure 14) as described in the previous paragraph.
7. Tune the feedforward portion of the control loop (block $G_{FF}(s)$ in Figure 14).

Comment to previous item:

- Comment to point 4 – generally data collection requires that some steps are performed on the excitation signal. Very often it is impossible to directly change the value of Disturbance signal, because this is often a process signal manipulated from complete different process equipment. Therefore very often it is necessary to find a period of time, in which Disturbance signal changes without any manual intervention and with enough dynamic excitation (from the power spectrum point of view).

- Comment to point 4 and 5 – the task of point 5 is to identify the transfer function $G_d(s)$ (Figure 14). For this purpose the data collection (specified at point 4) should be performed taking the feedback controller in manual mode (open loop). If data collection is performed with feedback controller in automatic mode (close loop), it is difficult to separate the effect on PV due to Disturbance from the effect due to Feedback controller, and the bad model can affect heavily the final tuning parameters and then the tuning performances. In case the Process Engineers don't like to carry out data collection with the controller in manual mode, the suggestion is, if possible, to reduce the gain of the PID loop K_P up to 10 times (this suggestion is not applicable to integral loop); this allows to reduce the effect of PID controller showing the effect of Disturbance on PV in a more clear way.

ABB. All rights reserved

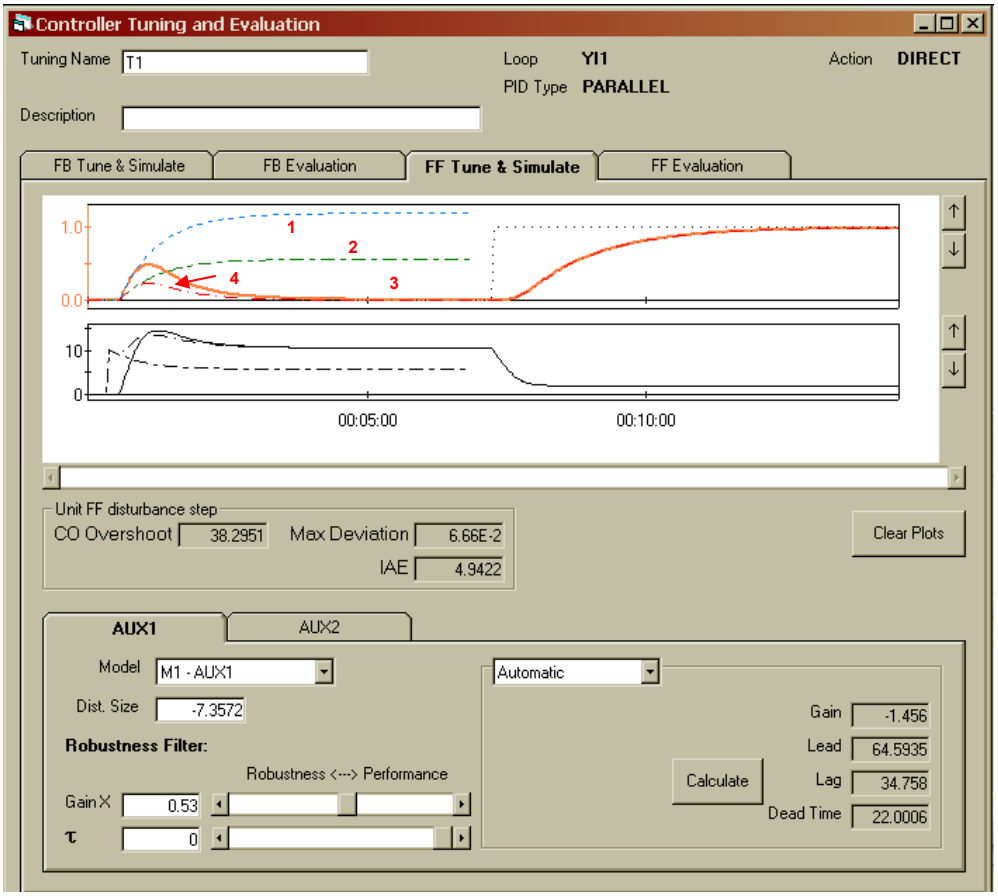


Figure 15 - FF Tune & Simulate

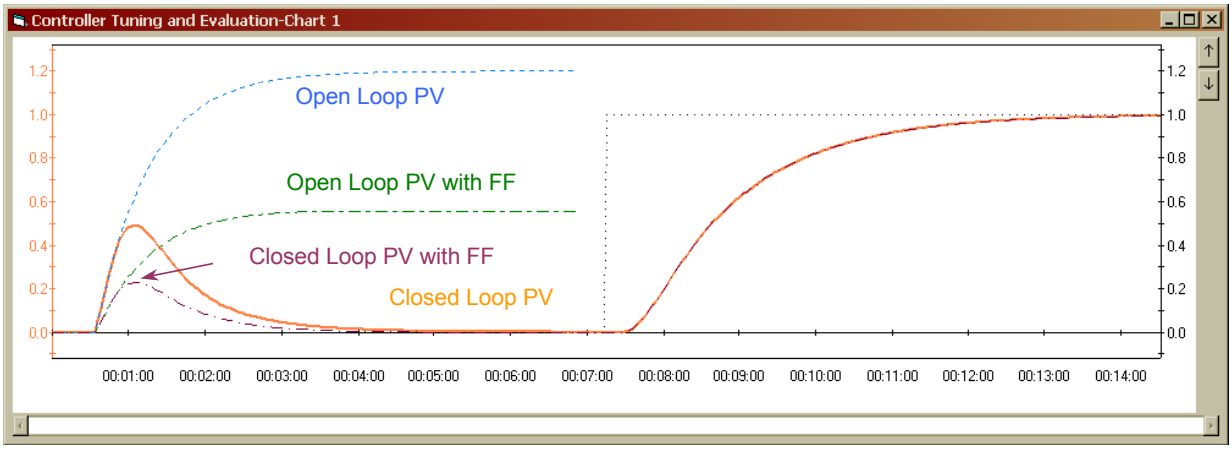


Figure 16 - FF Simulate Details

Figure 15 and Figure 16 shows the user interface for tuning feedforward loops. The FF tune and simulate tab does not appear unless there are FF models created and visible on the object tree. FF models are models that represent the effect of the Disturbance variable on the Process Variable (PV). For instance M1-AUX1 (in Figure 15) is a model associated to an Auxiliary Variable linked with a signal representing a Disturbance variable. These models are used in FF design

ABB. All rights reserved

The FF tune and Simulate tab will be inactive until the user selects tuning for the PID loop. This is because the effect of the PID control is taken into account in the FF simulations.

The auto-tuning method calculates the FF controller transfer function (gain, deadtime and time constants of the numerator and denominator terms). The deadtime of the FF controller is the difference between the Disturbance model deadtime and the process deadtime. If the Disturbance deadtime is shorter, there is no deadtime in the feedforward transfer function.

After the feedforward transfer function is calculated, LPM generates plots for a simulated step Disturbance in the FF variable (d). There are four traces in the plot:

1. Open loop response to FF Disturbance (without feedback and without FF)
2. Response to FF Disturbance with FF control (without feedback)
3. Closed loop response to the FF Disturbance with feedback control (without FF)
4. Closed loop response to FF Disturbance with feedback and FF control

These traces are numbered 1 through 4, respectively, in Figure 15, while they are explicitly labeled in Figure 16.

The FF Evaluation tab also includes 3 Bode plots that show the Magnitude of Transfer Function of Sensitivity to Disturbance Rejection. There are three traces in the plot:

- Sensitivity to Disturbance signal with Feedback control active (FF control inactive)
- Sensitivity to Disturbance signal with only FF Control active (Feedback control inactive)
- Sensitivity to Disturbance with feedback and FF control active

These diagrams show for each frequency the capability of control system to reduce/amplify the signal from Disturbance (input) to Process variable (output). The FF control action should reduce the magnitude of sensitivity to disturbances rejection transfer function. Typically FF controls are more effective at the intermediate frequency.

For manual tuning of a FF loop the user specifies the gain, deadtime and time constants for the FF controller and clicks Calculate to view the simulation and frequency response results.

2.4.2.6 Tuning Cascade Loops

Figure 17 displays a block diagram of a cascade control scheme. There are two PID controllers, one for the inner loop and one for the outer loop. The control output of the outer loop PID controller is the setpoint of the inner loop PID controller. The block diagram also includes disturbance inputs and models for the inner loop and outer loop dynamics (Gp Inner and Gp Outer, respectively).

ABB. All rights reserved

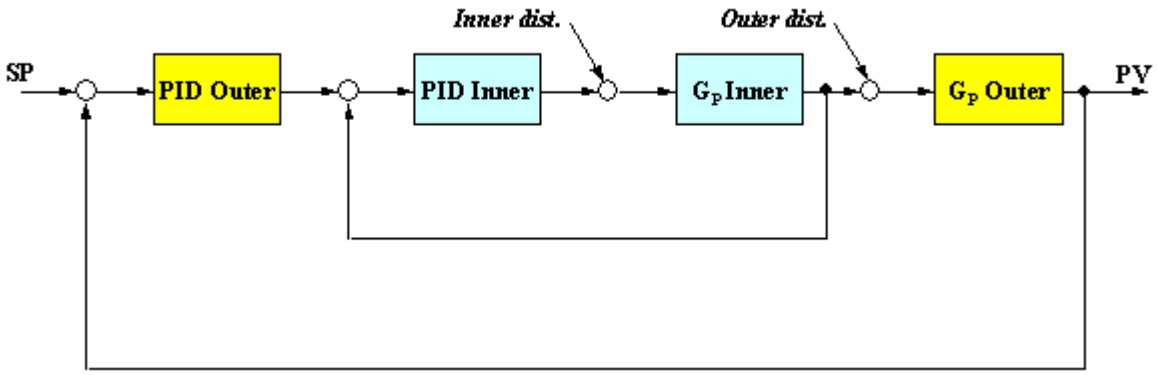


Figure 17 – Cascade Controller

In LPM, cascade controllers can be defined in one of two ways:

(i) Inner loop and outer loop are handled the same as any standard loop, or
(ii) Outer loop is handled in a special manner by associating it with an inner loop.

In method (i), the user configures and tunes the inner loop and outer loop separately. The inner loop is tuned first, then the outer loop. The inner loop and outer loop are each treated as a standard PID control loop. When collecting data for the outer loop, the inner loop PID controller should be in AUTO mode and the outer loop model is developed from its control output (CO) signal to the outer loop process value. The tuning for the outer loop also proceeds in a standard manner as for any other PID control loop.

For method (ii), the user first defines the configuration and tuning of the inner loop in the same manner as for method (i) (i.e. same as a standard control loop), but then the outer loop is configured and tuned in a special manner. In particular, the user should configure the outer loop as a Cascade Master (Outer) loop of a cascade control scheme and identify the name of the inner loop. *Figure 18 - Configuration form for Cascade Master / Outer loop shows the loop configuration form for loop "LC54".*

ABB. All rights reserved

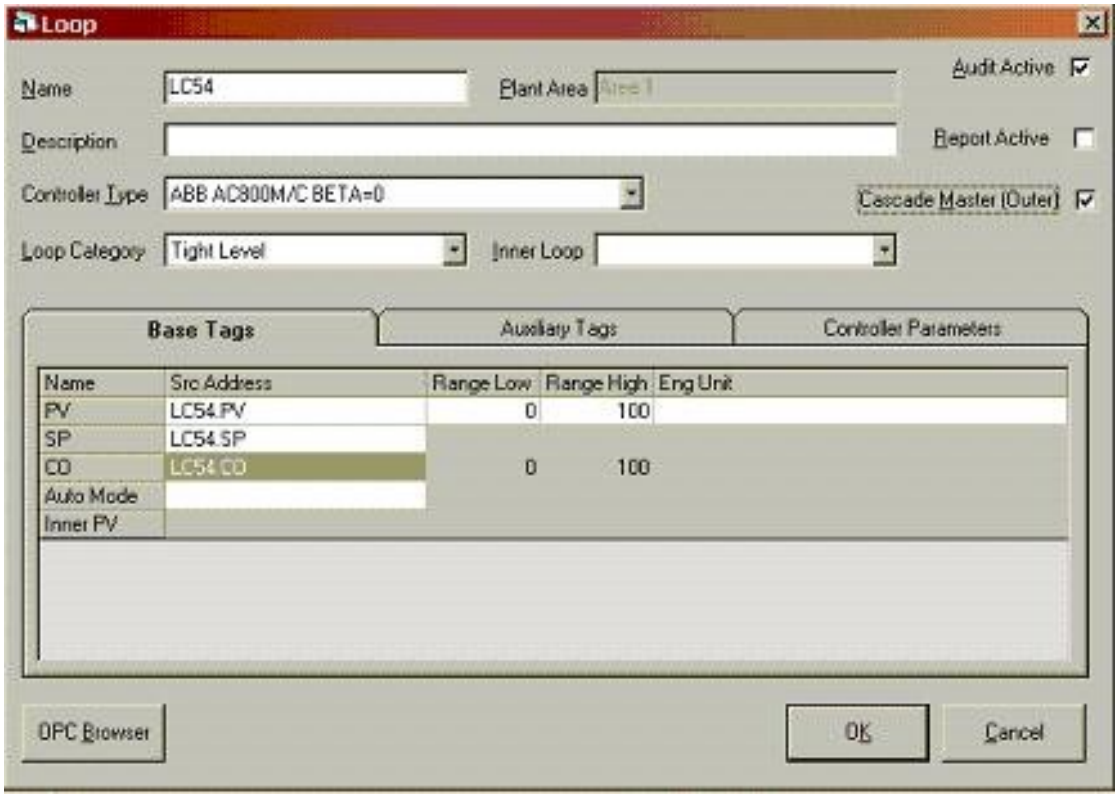


Figure 18 - Configuration form for Cascade Master / Outer loop

The user should click the checkbox labeled "Cascade Master (Outer)" to define it as an outer loop of a cascade control scheme. The user should then select the desired inner loop from the Inner Loop combo box, which lists all of the configured loops. Note that the Inner Loop combo box only appears if the Cascade Master check box is selected. Click on the OK button to save the loop configuration.

When a loop is configured as a cascade master / outer loop, any new modeling and tuning objects will be treated differently from those defined for a standard loop. For example, in a modeling object of an outer loop, it is assumed that the model is defined between **the inner loop PV and the outer loop PV** (i.e. corresponding to Gp Outer) instead of between the outer loop CO and the outer loop PV. The outer loop model should be created as follows:

- Note that the Src (Source) Address for the CO of the outer loop is not editable on the Base Tags tab of the loop configuration form. This Source address is automatically defined to be the same as the Source Address for the PV of the inner loop (to obtain the proper data for building the outer loop model).
- Create a new data collection object for the outer loop. Start the data collection and ensure that the inner loop PV is changing sufficiently (to build model of Gp Outer). Save the dataset.
- Create a new modeling object. Select the newly collected dataset, then choose appropriate modeling parameters and create a model. Save the new model. The new model will appear in the object tree with the extension "-CASC" (instead of "-CO") to indicate that it represents a cascade master / outer loop.

ABB. All rights reserved

2.4.3 Web Client

An important enhancement of LPM functions is provided by the possibility to interface and even interact with LPM from remote. This is achieved through the use of LPM Web Client, a facility that can be installed on a remote PC and is able to access any LPM located in PC connected over the same LAN and to perform the following actions:

1. Access existing Tuning Logs;
2. Access existing Auditing Reports:
3. Configure and launch new Auditing Reports.

Figure 19 - LPM Web Client shows a snapshot of the LPM Web Client HTML page.

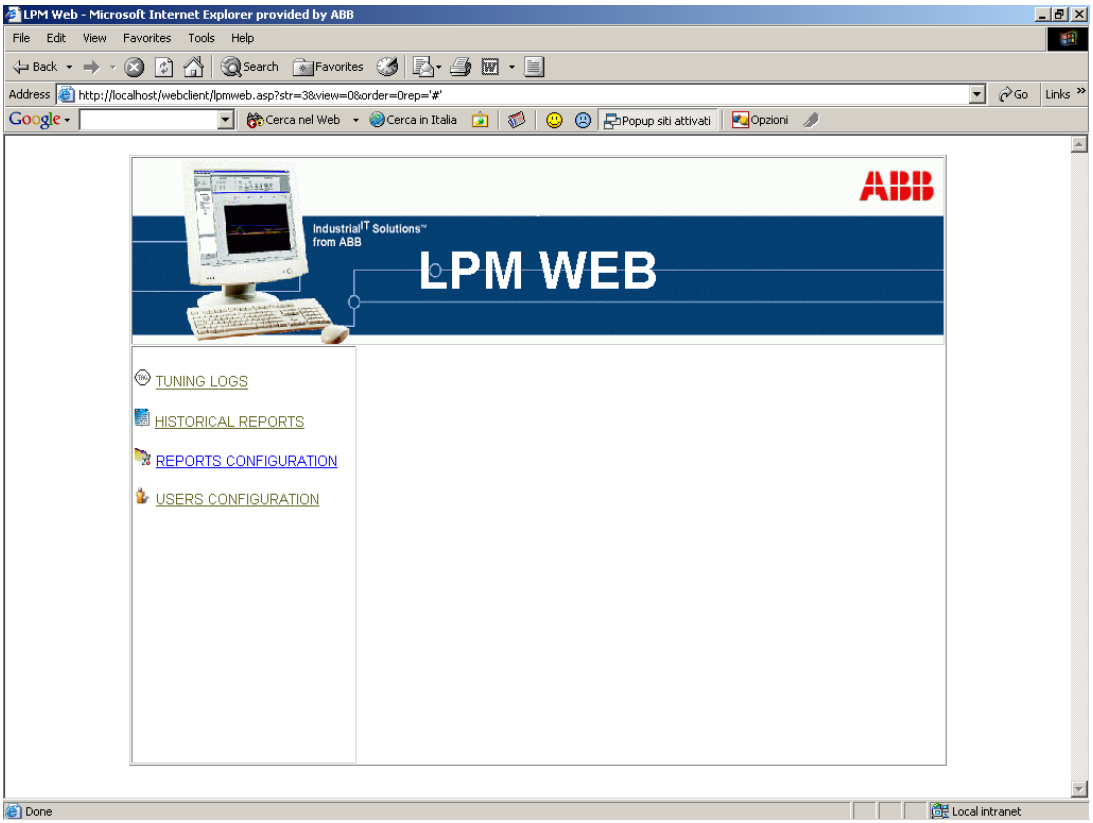


Figure 19 - LPM Web Client

ABB. All rights reserved

While LPM computes indices as real numbers, the diagnoses are evaluated as true or false (1 or 0). The diagnoses rules are natural language statements. The diagnoses are listed in the following.

- Acceptable Overall Performance
- Tuning Problem
- Loop Oscillatory
- Setpoint Oscillatory
- Significant External Disturbance
- Significant Non Linearity
- Valve Stiction
- Valve Leakage
- Valve Size Incorrect
- Excessive Valve Action
- Data Unreliable

Each diagnosis is computed from a set of simple logic statements using the performance indices. For instance the diagnosis for sluggish tuning is based on the autocorrelation index, error oscillation index, percentage of time saturated and setpoint crossing index. Some rules use only one index, while the overall performance diagnosis is comprehensive, using several performance indices and five of the hypotheses. ABB determined the appropriate rules through field tests and extensive analysis of process data.

Reports can also include a "What has changed" sheet where the indices affected by the wider variations from the previous month or week are highlighted for easy and prompt identification of arising problems.

2.4.3.1 What Auditing means to you

Close supervision of control system performance is a cost effective way to improve process economics and prevent degradation over time. Auditing control loops delivers the following types of benefits:

- Extended equipment lifetime
- Improved energy efficiency
- Increased process capacity
- Fewer plant trips
- Higher and more consistent product quality

Plant auditing addresses the core asset management goals of optimum performance and maximum return on investment. Audited plants attain stable and robust conditions necessary for higher-level optimization functions offered by model based multi-variable controllers and/or true plant optimizers.

ABB. All rights reserved

3. Installing LPM

3.1 *How to install LPM*

Insert the LPM CD in your CD-ROM drive, the LPM Setup will automatically start. If the LPM Setup does not automatically start, click on *Start* button on the Windows taskbar, select *Run...* type ***D:\Loop Performance Manager.msi*** (change the CD-ROM driver letter according to the personal computer configuration) and then click **OK** (see Figure 20).

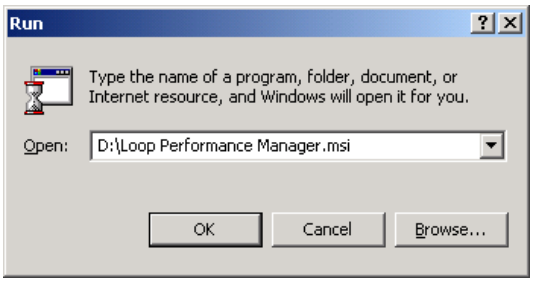


Figure 20 - Dialog Box to Install LPM

Setup cannot install system files while they are in use thus, before proceeding with the installation, you should make sure to close any open application.

As first step of the installation, the ***LPM-License Agreement*** is presented. In order to proceed with the setup, the license agreement has to be accepted. This is accomplished by selecting **Accept License Agreement** in the bottom left side of the form and then by pressing the **OK** button.

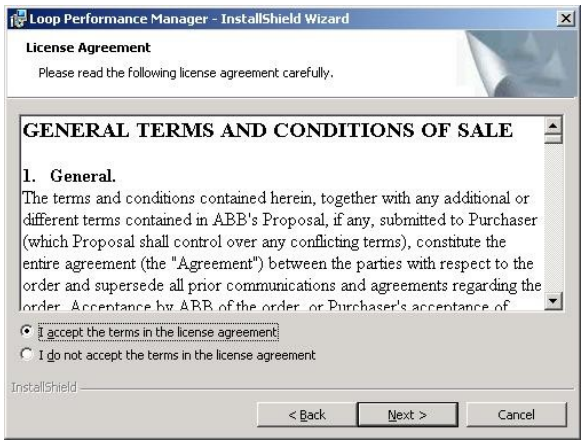

Loop Performance Manager - InstallShield Wizard
License Agreement
Please read the following license agreement carefully.
GENERAL TERMS AND CONDITIONS OF SALE
1. General.
The terms and conditions contained herein, together with any additional or different terms contained in ABB's Proposal, if any, submitted to Purchaser (which Proposal shall control over any conflicting terms), constitute the entire agreement (the "Agreement") between the parties with respect to the order and supersede all prior communications and agreements regarding the
I accept the terms in the license agreement
I do not accept the terms in the license agreement
InstallShield
< Back
Next >
Cancel

Figure 21 - Licensing Agreement Accept Dialog

The next steps of the installation procedure take a few minutes.

ABB. All rights reserved

The Setup program installs LPM executable files and a Program Group is created as ABB Industrial IT\Optimize IT\Loop Performance Manager. At the end of the installation, a Dialog Box notifies the user that the setup has been completed successfully.

3.2 *Installation Summary*

To install Loop Performance Manager:

1. Log in as a user with administrative privileges.
2. Quit all other running programs.
3. Insert the Loop Performance Manager installation CD into the CD drive.
4. If the Loop Performance Manager.msi program does not start automatically, invoke the installation program. To do this, either:
   a. Select:

      Start > Run > Browse

      and locate and select Loop Performance Manager.msi on the Loop Performance Manager installation CD. Click **Open** and then **OK** to begin the installation.

      - or -

   b. Use Windows Explorer to locate **Loop Performance Manager.msi** on the Loop Performance Manager installation CD. Double click **Loop Performance Manager.msi** to begin the installation.
2. Read the information and follow the instructions provided in the **Product info:** box.
3. Click **Install** and follow the instructions provided by the installer.
4. At the appropriate point in the installation procedure, choose a mode of installation (**Complete**, or **Custom**). You can always modify your installation later if you discover that you need some other software component or want to delete some superfluous component.
5. Let the installation procedure finish.
6. Restart the workstation, if requested.
7. Install **License Entry** if it is not already installed. It is also necessary to add the licenses.

3.3 *Post installation instructions*

After the installation procedure successfully ends, the user must perform a few additional steps in order to allow the LPM software to communicate with the data source(s).
All the above steps are intended with the goal to let any LPM's module run in a security context having Administrative rights on the Data Source LPM will use to collect data from.

<u>Notice</u> The following operations must be executed with LPM Service and LPM Client not running (stopped). If these are executed with LPM Service and LPM Client running, you need to re-boot the machine to make these changes effective.

ABB. All rights reserved

In order to stop the service click on Settings/Control Panel from your START Menu, select Administrative Tools (if working under Windows 2000 or higher) and double click on the Services icon. Locate and select LPM Auditing Server and click on the Stop button on the toolbar.

Server Configuration (PC running OPC Server)

- Create a User with administrative rights (or choose an existing one having same requisites) that will be used by LPM for client/server communication with data source.
- Launch DCOMCNFG.exe from dos prompt. (See Figure 27)

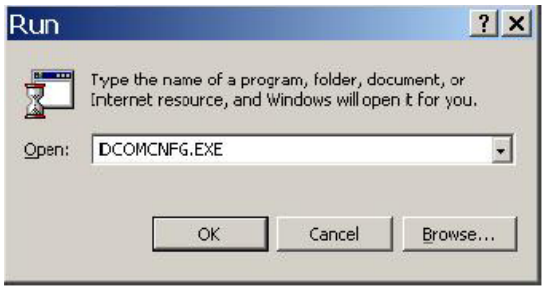


Figure 22 - DCOMCNFG Dialogue

- A form like the one shown in Figure 28 will appear:

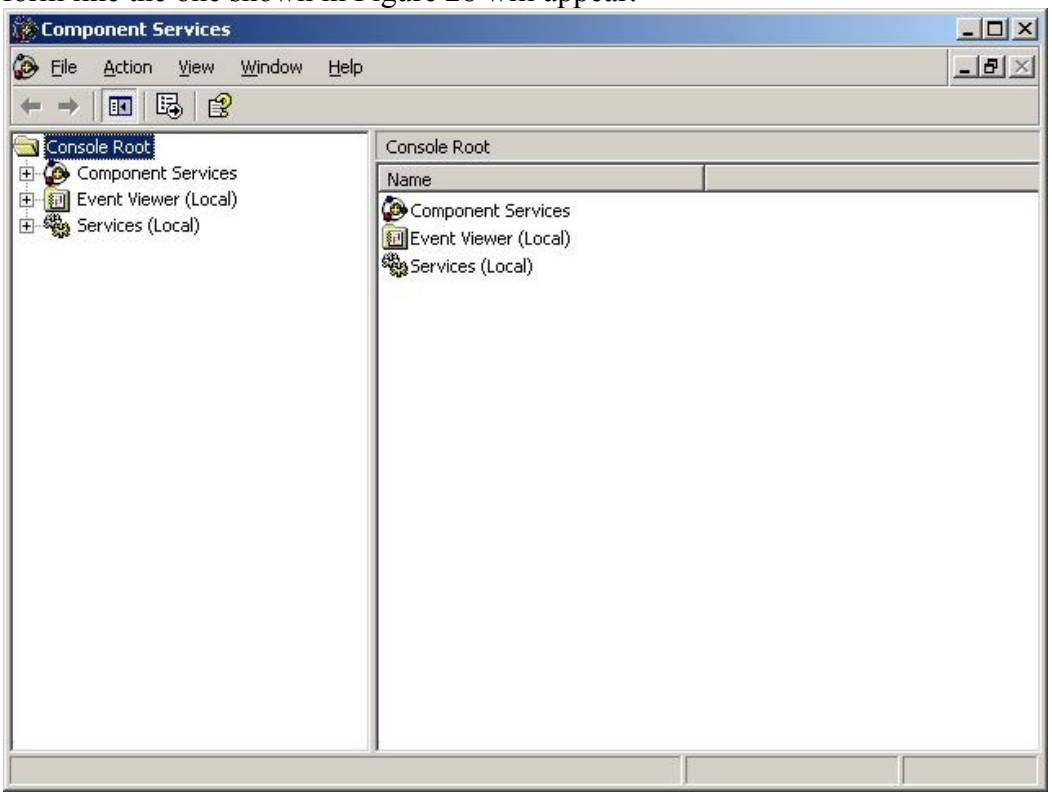


Figure 23 - DCOM Configuration

ABB. All rights reserved

- Double click on “Component Service” → “Computers”.

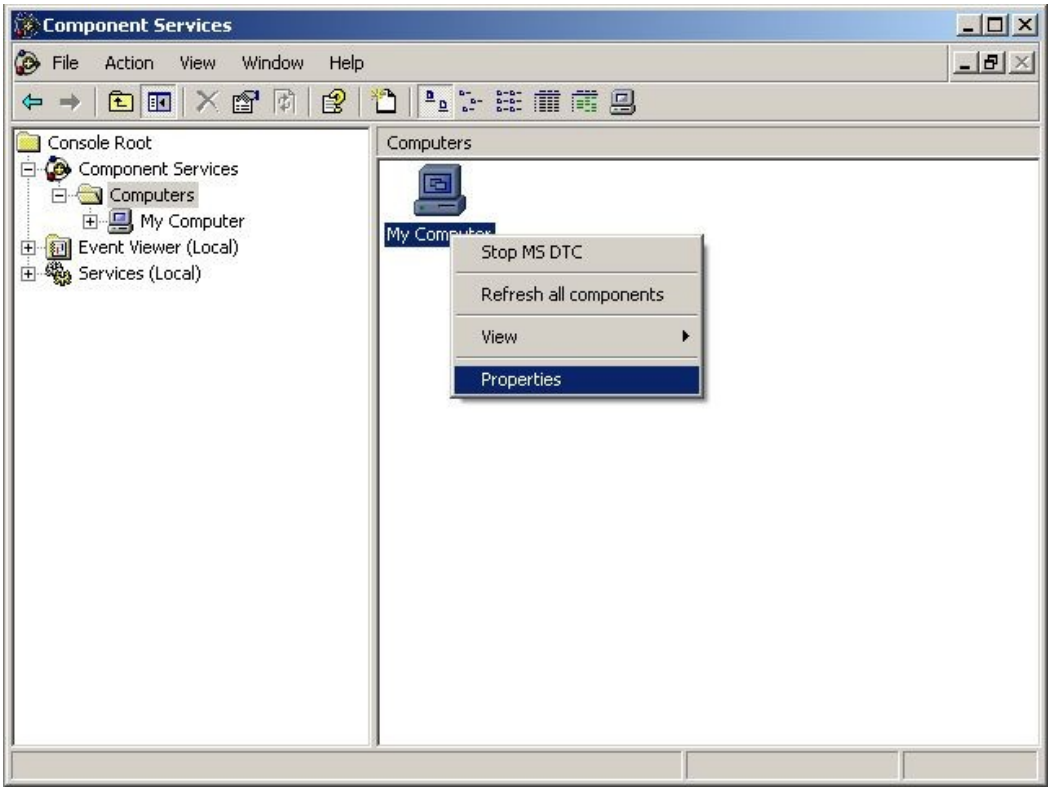


Figure 24 - DCOM Configuration

- Right click on “My Computer” and select Properties. Go to “COM Security” tab.

ABB. All rights reserved

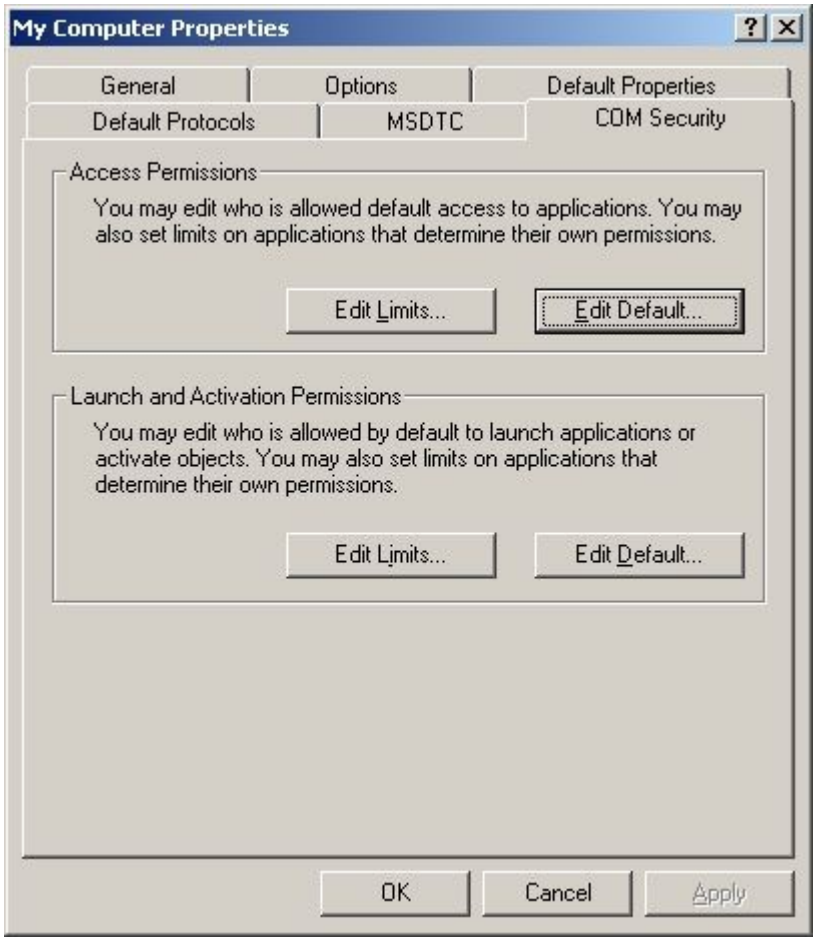

My Computer Properties
General
Options
Default Properties
Default Protocols
MSDTC
COM Security
Access Permissions
You may edit who is allowed default access to applications. You may also set limits on applications that determine their own permissions.
Edit Limits...
Edit Default...
Launch and Activation Permissions
You may edit who is allowed by default to launch applications or activate objects. You may also set limits on applications that determine their own permissions.
Edit Limits...
Edit Default...
OK
Cancel
Apply

Figure 25 - DCOM Configuration

- Click “Edit Default” button in the frame “Access Permission” and “Launch and Activation Permission”; ensure that the Administrators user group is in the list. And set the check box like image (all selected).

ABB. All rights reserved

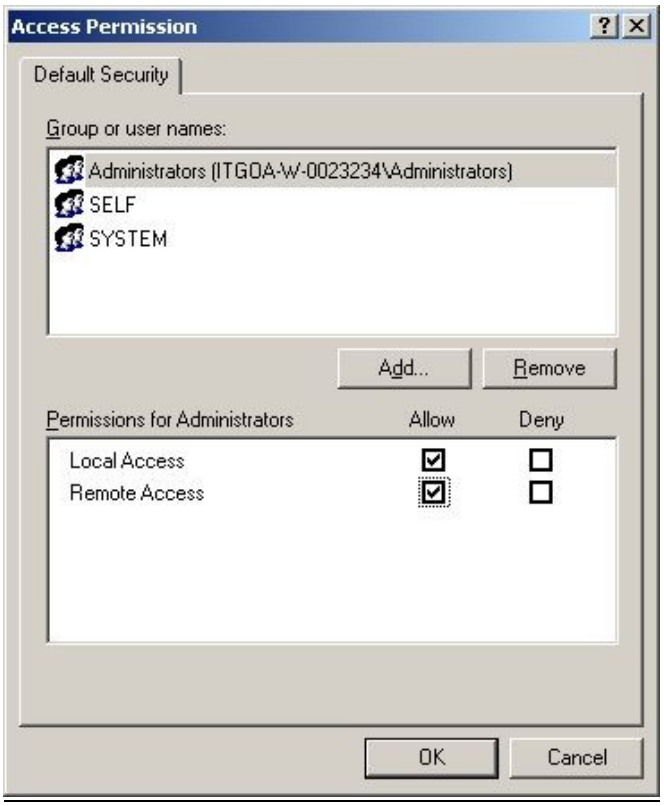


Figure 26 - DCOM Configuration

ABB. All rights reserved

Client Configuration (PC running LPM)

- Open DCOMCNFG on client (cpmPlus LPM) machine. Double click on "My Computer" → "DCOM Config".

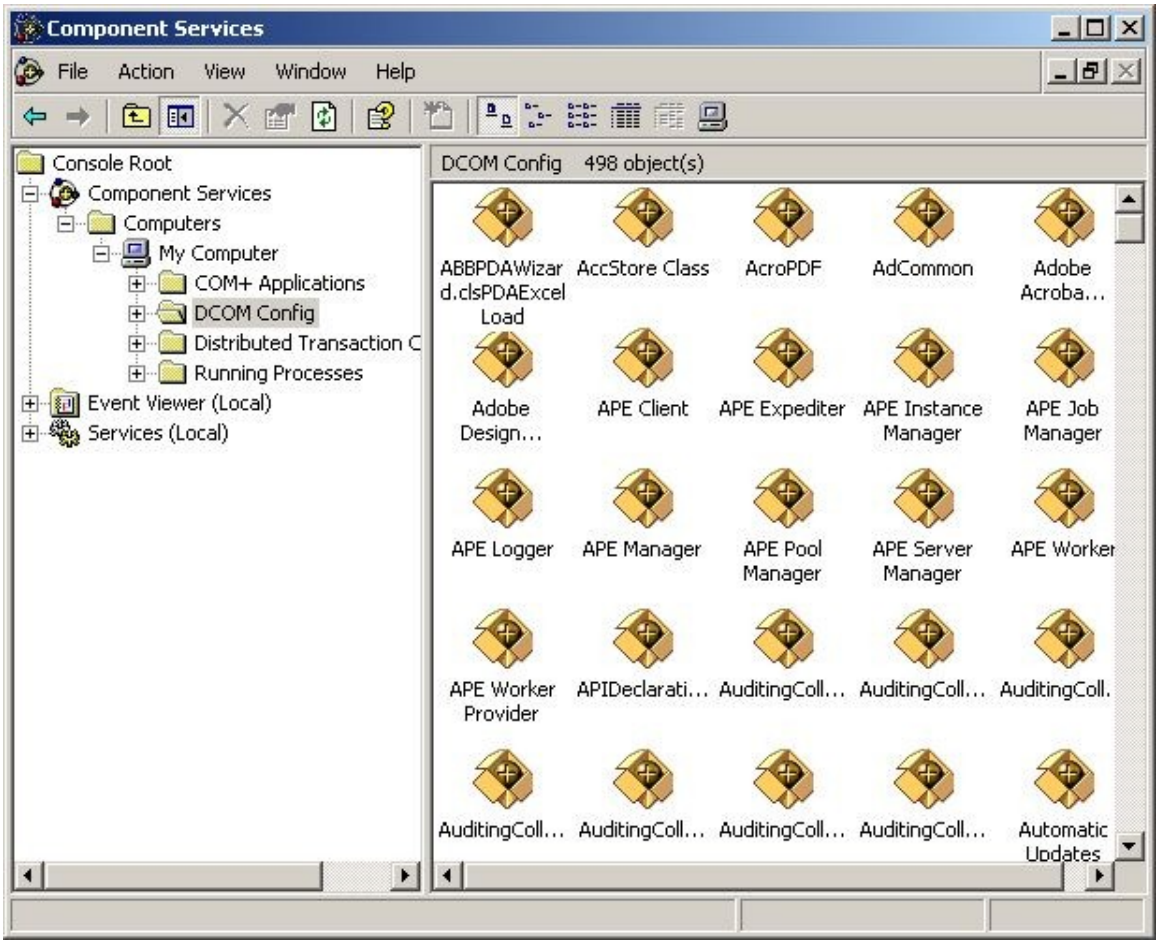


Figure 27 - DCOM Configuration

For any entry starting with **AuditingCollection**, **DataCollection** (only if you login on PC with an user that have no access permission on OPC machine), **LPMServerEXE** and **LPMRepServer**:

Select the correct entry and press "Properties" button.

In Security tab:

- Check radio button "Use custom Access permission".
  Click on "Edit" and Add Everyone – type of access "Allow Access"

- Check radio button "Use custom Launch permission".
  Click on "Edit" and Add Everyone – type of access "Allow Launch"

- Check radio button "Use custom Configuration permission".
  Click on "Edit" and Add Everyone – type of access "Full Control"

ABB. All rights reserved

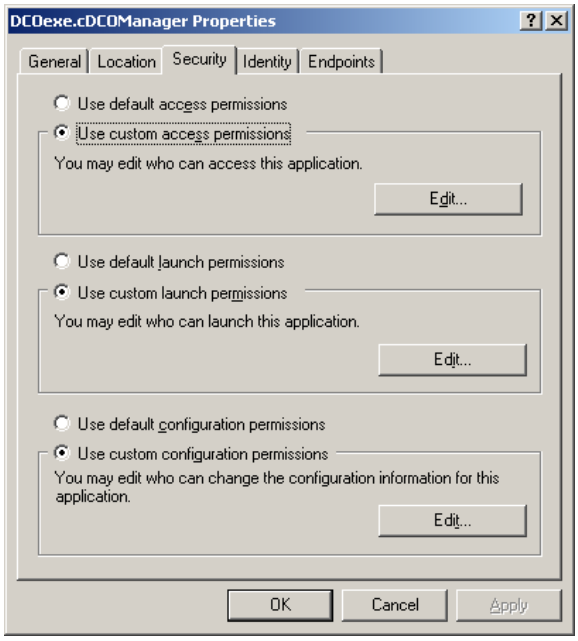


Figure 28 - Security Tab

In Identity Tab:

- Check "This User" and set credentials of a user with Administrative rights on OPC server. (The one created in section **Server Configuration** at first point)
  In case there is no distinction between Client and Server PC (I.e. LPM has been installed on the same PC where data Source is located) specify credential of user used to log-on the Operating System, ensuring that such user is included in Administrators group of the local PC.

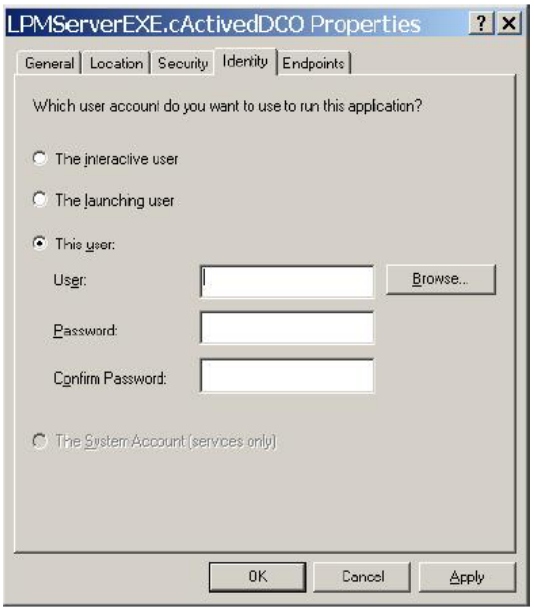


Figure 29 - User Identity

ABB. All rights reserved

3.4 *Software Licensing*

The software licensing procedures are necessary for proper operation of the ABB software products and features that require licenses. Software is licensed by relating a software key for each software feature with a unique machine ID. Software keys are obtained from the ABB software factory system.
This section describes the procedures for obtaining machine IDs, requesting software keys, and installing software keys. Additionally, the procedures are provided to install and run the license entry program.

An overview of the software licensing procedure is shown in Figure 30 .

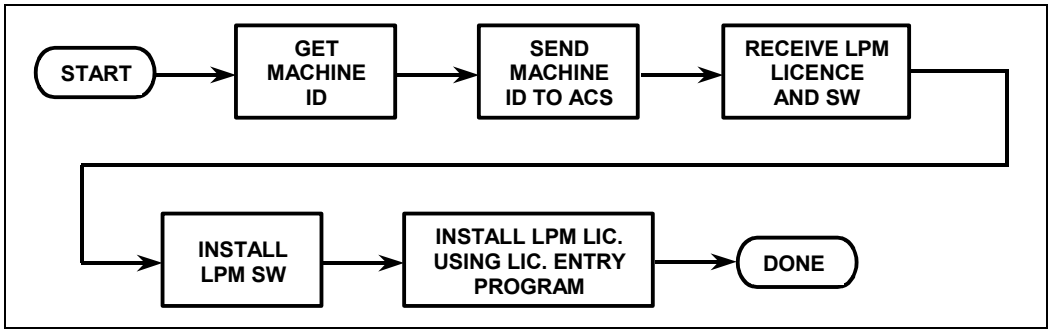


Figure 30 - Software Licensing Procedure

Licenses are generated for the machine ID of each workstation. The machine ID identifies the workstation where the ABB software will run, and the software periodically checks that the machine ID is still present. A single machine ID supports multiple software keys for different features and for different applications.

3.4.1 License Entry Program

The license entry program is used to activate the software features once the necessary software keys have been installed.

3.4.1.1 Requirements

The license entry program can be installed and run on a Windows 2000 (and later versions) workstation.

3.4.1.2 Install

The license entry program is installed with the software in the folder:

C:\ Program Files\ ABB Industrial IT\ Optimize IT\ Loop Performance Manager\BIN

3.4.1.3 Uninstall

This program uses the standard Microsoft uninstall procedure.

1. Open Control Panel by selecting:

   Start > Settings > Control Panel

2. Double click **Add/Remove Programs** and the Add/Remove Programs window appears.

ABB. All rights reserved

3. Select **ABB License Entry Installation**.
4. Click **Change/Remove**.
5. A window appears asking if you are certain you want to uninstall the specified product. Click **Yes** if you are sure, otherwise click **No**.
6. Wait for the uninstall procedure to finish.

3.4.1.4 Running the License Entry Program

After installation, the license entry program can be invoked from Windows. By selecting:

Start > Programs > ABB Automation > License Entry

3.4.2 Selecting a Machine ID

LPM license is based on the Ethernet™ address of the PC, where LPM has to be installed. If the Ethernet card is replaced or permanently removed from the workstation, it will be necessary to reapply for the software license.

The recommended method for determining the machine ID is as follows:

1. Select the **Available IDs** tab of the license entry program (Figure 31 - Available IDs Tab).

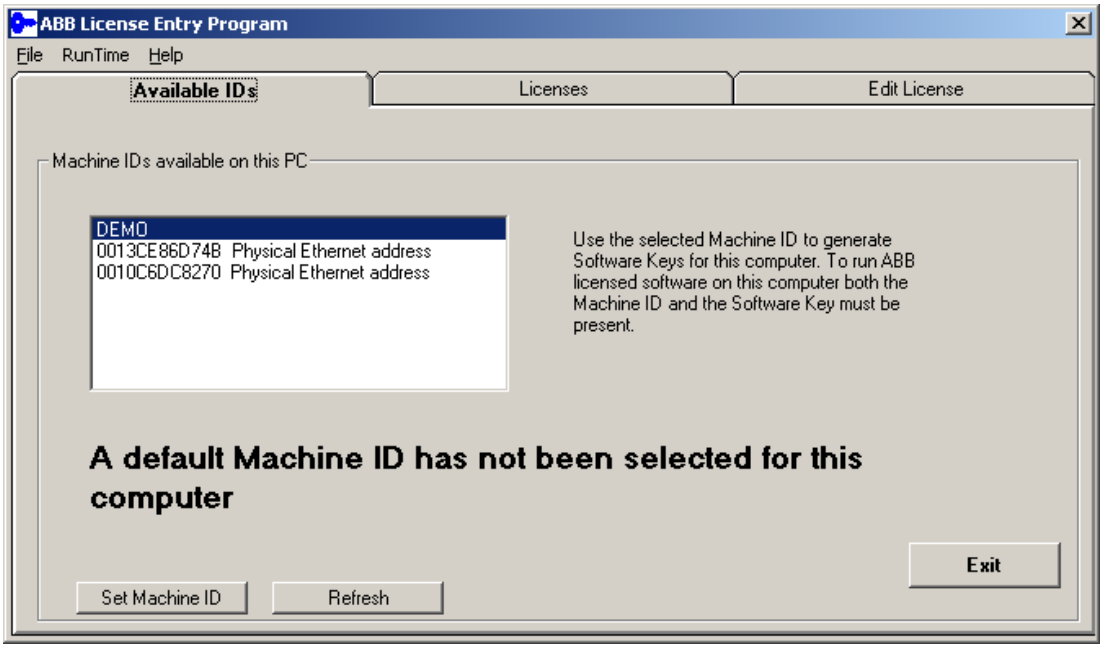


ABB License Entry Program
File RunTime Help
Available IDs
Licenses
Edit License
Machine IDs available on this PC
DEMO
0013CE86D74B Physical Ethernet address
0010C6DC8270 Physical Ethernet address
Use the selected Machine ID to generate Software Keys for this computer. To run ABB licensed software on this computer both the Machine ID and the Software Key must be present.
A default Machine ID has not been selected for this computer
Exit
Set Machine ID
Refresh

Figure 31 - Available IDs Tab

2. Click **Refresh** to update the list of machine IDs. The choices for machine IDs are:
   - Demo.
   - Ethernet address.

ABB. All rights reserved

3. Select one of the choices and click **Set Machine ID**. This causes the selection to be the default used in other tabs.

An alternative method is to use the following procedure to obtain the Ethernet card address. The disadvantage of this method is that it can report incorrect values if it is used under emulation systems like Exceed.

1. From the command prompt, type **ipconfig/all Enter**. The Ethernet address can be obtained by choosing any listed Ethernet adapter and recording the physical address line item. For example:

   00-60-08-CC-BB-B8

2. Record this machine ID (without the hyphens) on the ABB licensing registration form and refer to Requesting Software Keys to request software licenses from ABB.

3.4.3 Requesting Software Keys

The following paragraphs describe the steps necessary to request software keys.

1. Refer to Selecting a Machine ID to obtain the machine ID of the PC on which the software is installed or on which it will be installed.
2. Locate and have available the registration number and the validation number on the registration form shipped with the software to be registered.
3. Complete the registration form provided.

- or -

Generate and complete a new software key request form using the license entry program. To do so:

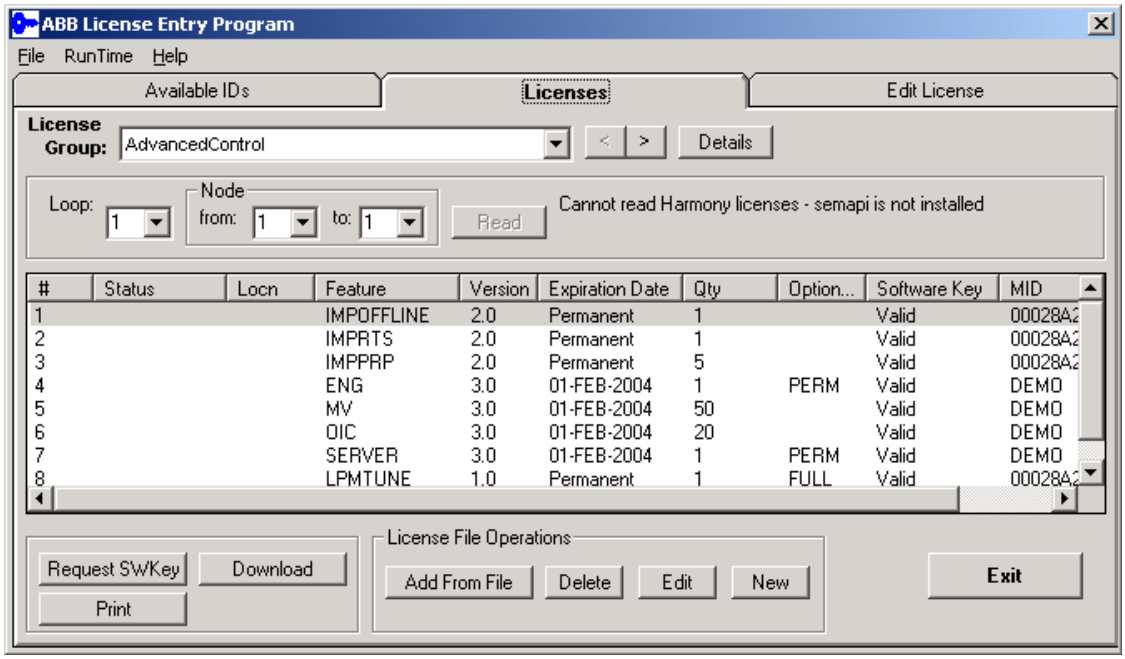


Figure 32 - Licenses Tab

ABB. All rights reserved

1. Select the Licenses tab of the license entry program (Figure 32 - Licenses Tab).
    a. Select: File > Add License Group. The ABB License Entry - Add License Group dialog box appears.
2. Enter AdvancedControl and click OK.
    a. A standard Windows Browse dialog opens. Navigate to the desired location, specify a file name, and click **Open**. For example, C:/Program Files/.../license.sla.
3. A message box appears asking: Are licenses for this group downloaded to a Harmony device? Click **No**.
4. Click **Request SWKey**.
    a. Verify that the machine ID is correct. If not, enter a new machine ID.
    b. Click **OK** to continue.
    c. Enter the registration number sent with the software and click **OK**.
    d. Enter the validation number sent with the software and click **OK**.
    e. Edit the form as necessary, and save it to a file or print it.
5. Send the registration form to the ABB software license administrator using one of the methods described on the registration form.
6. Wait for the software keys to be returned. The method by which the software keys will be returned is selected on the registration form. The choices are Internet mail, postal mail, or fax. A mail address or fax number must be included on the form.

3.4.4 Entering Licenses

Licenses must be entered on the appropriate workstation in order for the license server and the applications to find them and run correctly. The form to request license keys should indicate which machine IDs the keys are to be registered against. Depending on the LPM configuration chosen you may need to install one of both of the following licenses:

Feature name	Features covered by license
LPMTUNING.	Tuning
LPMAUDITING.	Auditing
LPMPDA	Plant-wide disturbance analysis[1]
LPMUtility	Taglist creator utility for AC800F and Infi90
LPMWEB	Web client

Table 3 – cpmPlus LPM licenses

To correctly install the software keys:

1. Run the license entry program by selecting:

[1] Further described in the LPM PDA Manual.

ABB. All rights reserved

Start > Programs > ABB Automation > License Entry

2. Select the **Licenses** tab, see Figure 33 -Edit License Tab.
3. Select the applicable license group for the product or feature being licensed.
4. Each feature of the software requires a software key to function properly. Select a feature and click **Edit License**. This opens the **Edit License** tab.
5. Enter the following license parameters:
   - Version.
   - Expiration date.
   - Licensed quantity.
   - Software key.
   - Option string.

   The option string must be entered if it is provided. (The option string is designated as User String when the software key is sent.)

   The license parameters can be found on the software keys form received from the ABB software factory system. Verify that the parameters match the machine ID shown in the **License** tab. Figure 33 -Edit License Tab shows an example screen with the licensing information entered.
6. Click **Validate** to verify that the information entered matches the software key. Correct any errors found, including failing to enter the option string if it was provided.
7. Click **Save** to save the entered license information. This saves the information locally on the workstation.

   Be sure to click **Save** after entering and validating the information for each feature of the software. Clicking **Validate** will only validate the information that was entered, not save it.
8. Choose another feature, then repeat Step 5 through Step 7.

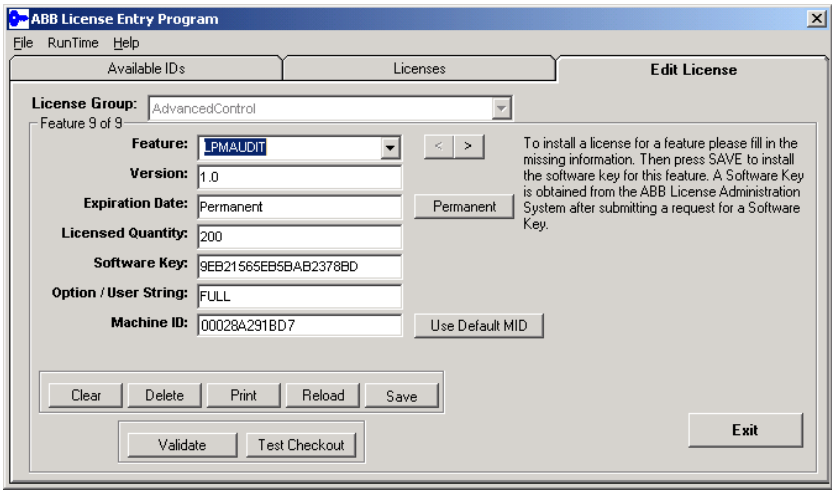


Figure 33 -Edit License Tab

ABB. All rights reserved

3.5 *Startup and Shutdown Procedures*

Loop Performance Manager is implemented as a set of Windows applications. LPM Client needs to be launched by the user. The underlying LPM Server can be automatically started as a service, when Windows is started[2].

Loop Performance Manager server can be shut down by the user.

To shutdown the LPM Server:

1. Locate the Windows Services window (on Windows 2000 operating system select **Start > Control Panel > Administrative Tools > Services**).
2. Select the LPM Server service and push the **Stop** button in the toolbar. .

To stop the LPM client:

1. Close the window (or select **File > Exit**).
2. Click **Yes** to confirm the action.

3.6 *Validation string*

A validation string is required to activate cpmPlus Loop Performance Manager. The validation string is provided together with the software license. Entry of the validation string is executed in LPM software and not in the license entry program.

The validation string is entered by using the Tools/Options menu entry. LPM operates in demo mode[3] only until the validation string is entered.

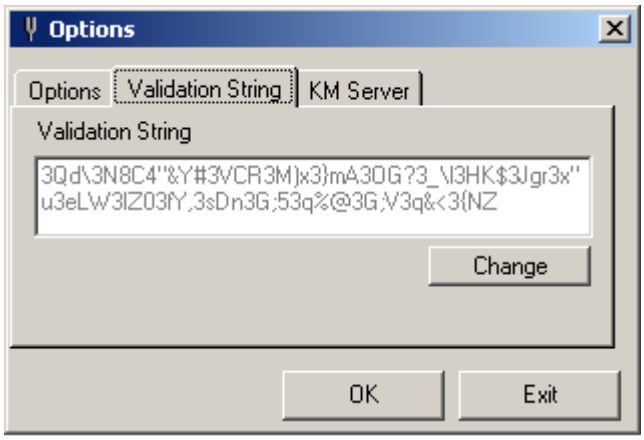


Figure 34 – Entering the validation string

[2] Automatic activation of LPM service should be enabled by the user.

[3] No save is allowed i.e. LPM project changes cannot be stored.

ABB. All rights reserved

4. Configuration/Application Building

4.1 *Auditing Workflow*

This section describes schematically the steps needed to properly monitor and assess control loop performances with CpmPlus Loop Performance Manager. Building an auditing application with LPM follows the workflow outlined in Figure 35 – Loop Auditing Workflow

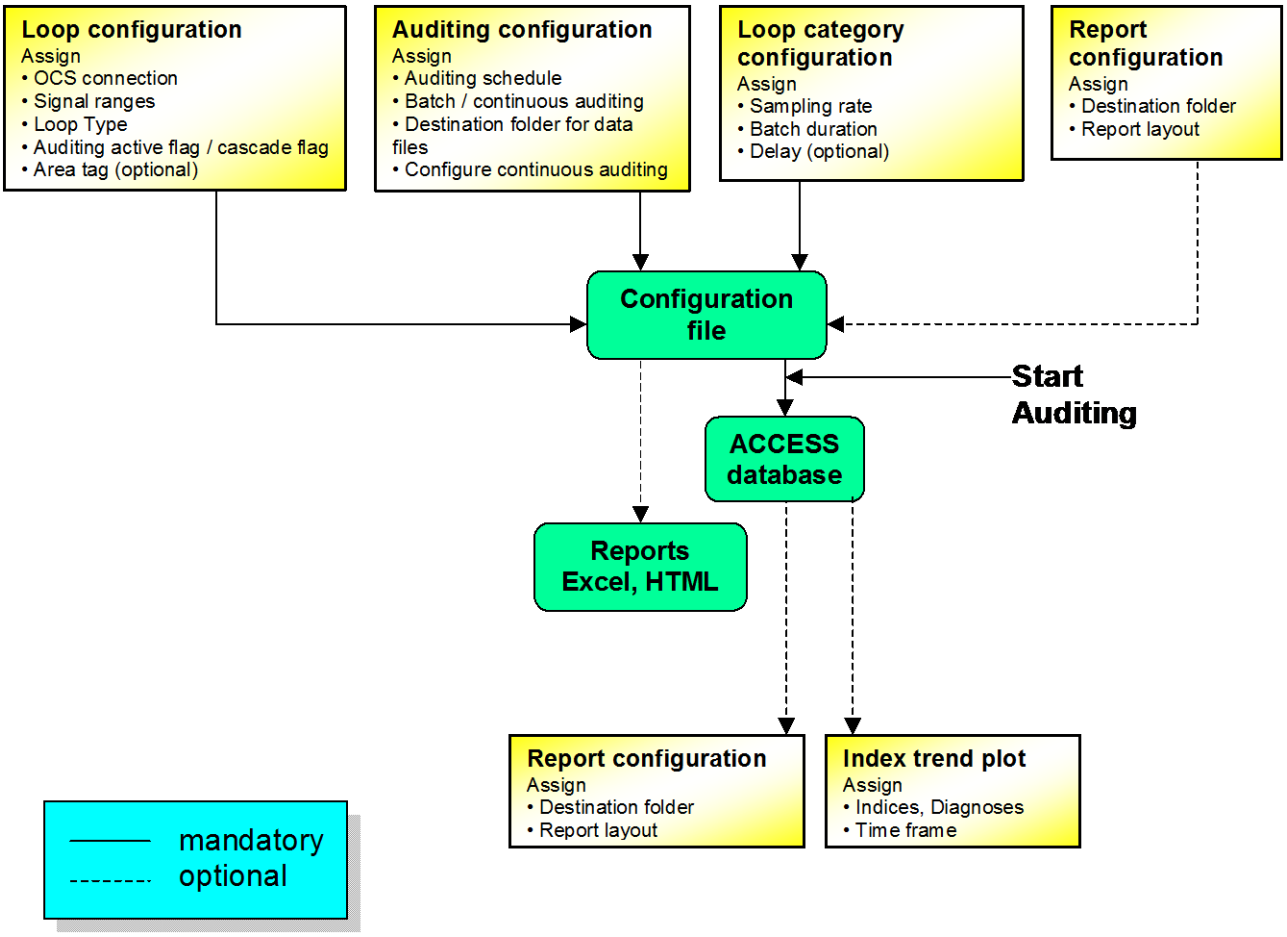


Figure 35 – Loop Auditing Workflow

In the next paragraphs a brief description of the common LPM GUI elements is given.
The detailed guide to each and every function available in LPM is provided in Chapter 6 (about Tuning) and 7 (about Auditing).

ABB. All rights reserved

4.2 *Tuning Workflow*

This section describes schematically the steps needed to obtain optimal PI/PID tuning parameters with CpmPlus Loop Performance Manager. Building a tuning application with LPM follows the workflow outlined in Figure 36 – Loop Tuning Workflow

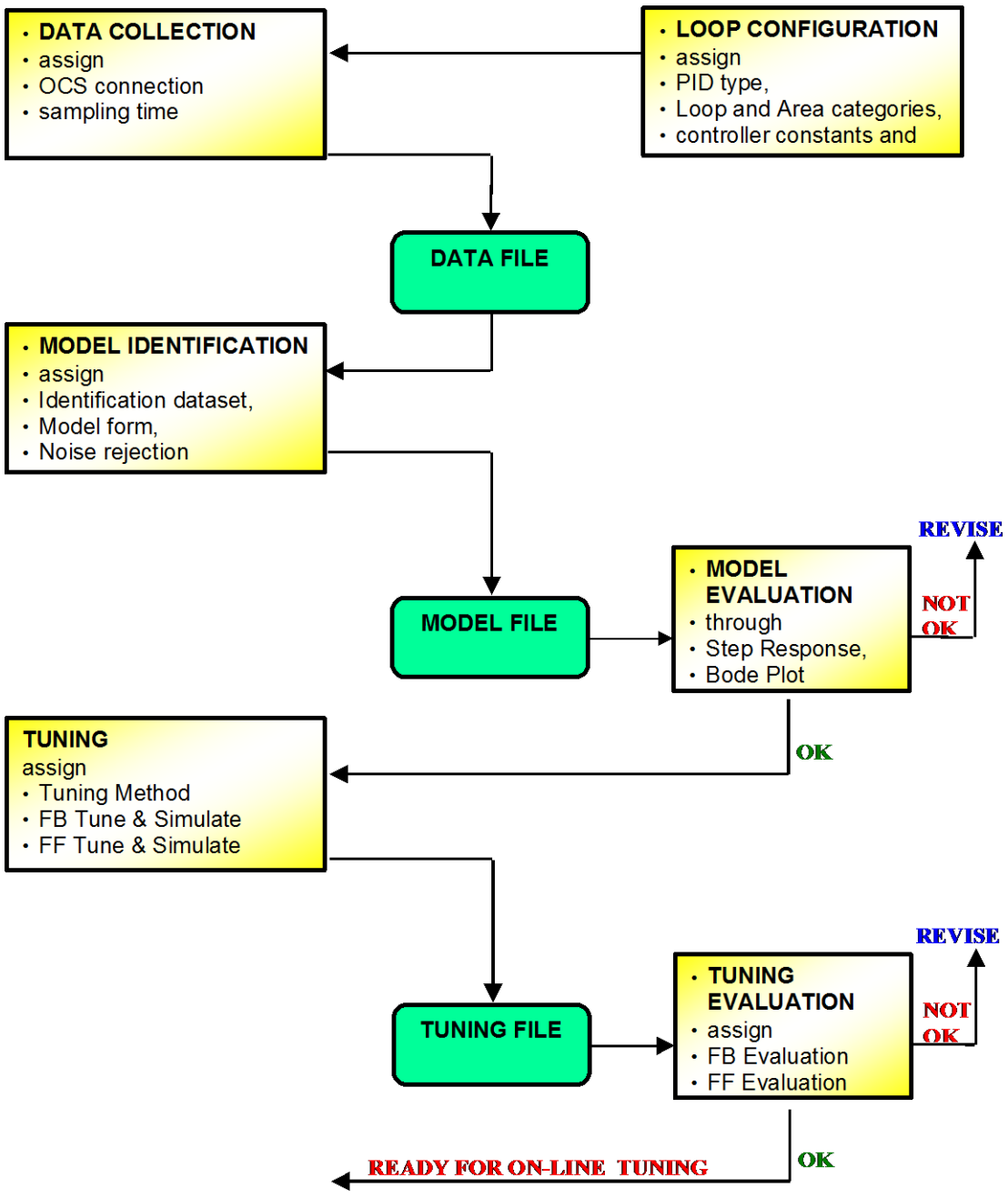


Figure 36 – Loop Tuning Workflow

Control system design is in many ways an iterative process, which is illustrated by the revise directives in Figure 36 – Loop Tuning Workflow. The user is assumed to have knowledge of necessary revisions when models or tuning specifications are not satisfactory.

ABB. All rights reserved

4.3 *LPM Client Tool Overview*

The Loop Performance Manager Client Tool is started from the **Start** menu. Select:
Start > ABB Industrial IT > Optimize IT > Loop Performance Manager > Loop Monitoring & Tuning

The main LPM Client Tool window appears as in Figure 37:

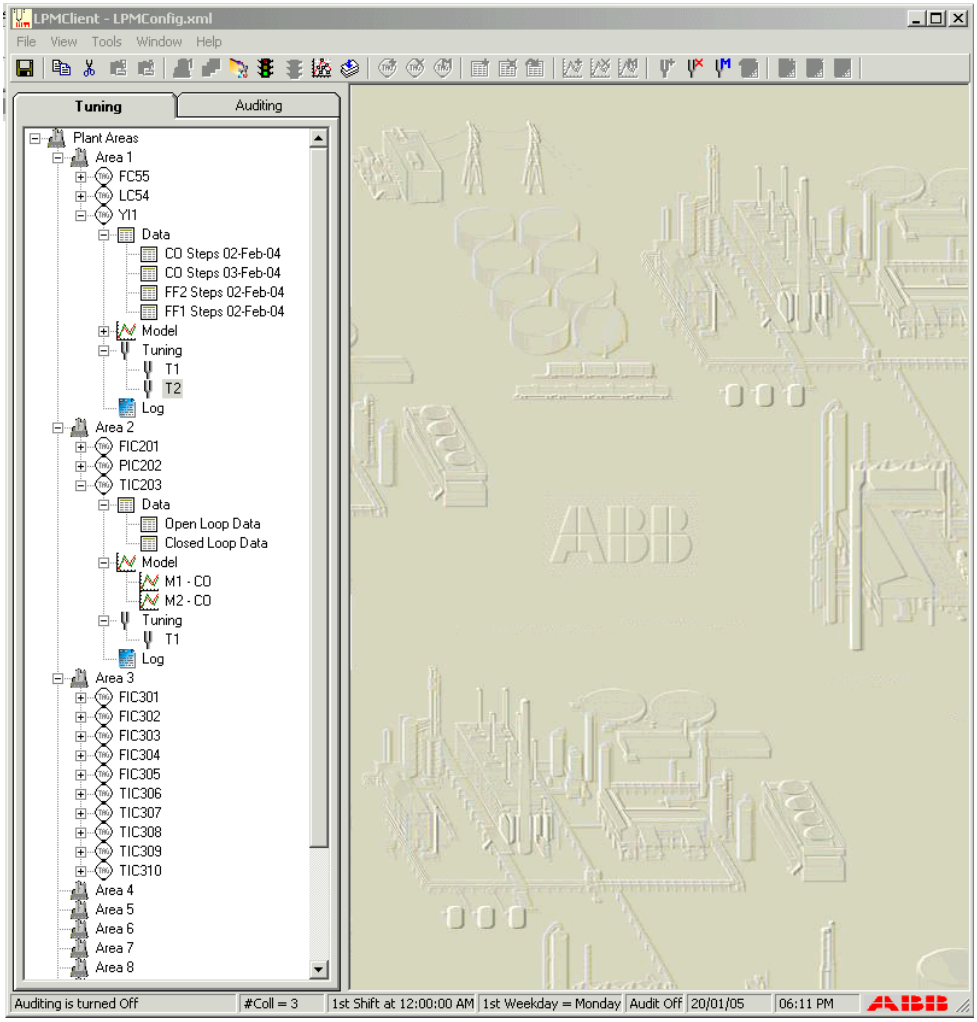


Figure 37 - Menus and Menu Items

The LPM Client Tool menus and menu items are listed in Table 4.

ABB. All rights reserved

Menu	Menu Item	Menu Item
File	*Save*	Saves configuration data.
	Exit	Exits the LPM Client Tool. The user is prompted to save if unchanged data exists.
View	*Toolbar*	Toggles the visibility of the toolbar.
	Status Bar	Toggles the visibility of the status bar.
	Refresh	Refreshes the viewing screen, including the configuration tree.
Tools	*Find Loop*	Looks for loops in the existing configuration
	Configure Auditing	Makes available the Auditing Configuration form
	Perform Auditing	Switch on/off Auditing functions
	Plot Diagnosis	Makes available the Plot Diagnosis form
	Plot Indices	Makes available the Plot Indices form
	Configure Report	Makes available the Configure Report form
	Interactive Report	Allows to generate interactive reports
	Data inspect	Allows to inspect existing historical data
	Advanced auditing settings	Allows to modify advanced settings for auditing
	Configuration summary	Shows configuration summary data
	Bulk Configuration	Makes available the Bulk Configuration form
	Bulk export	Allows to export configuration details
	Monitor Server Status	Makes available the Monitor Server Status form
	Options	Allows to open the option/properties form
Window	*Cascade*	Cascade currently active windows.
	Tile Horizontal	Tile currently active windows horizontally.
	Tile Vertical	Tile currently active windows horizontally.
	Arrange Icons	Arrange icons along the bottom of the viewable screen.
Help	*Contents*	Information about Loop Performance Manager.
	User Guide	Shows LPM User Manual
	PDA User Guide	Shows LPM Plantwide Disturbance Analysis User Manual
	About	Information about Loop Performance Manager.

Table 4 – LPM Client Tool Menus and Menu Items

4.4 *Toolbar*

The toolbar includes the following icons/items:

- Save
- Copy
- Cut
- Configure Area

ABB. All rights reserved

- Configure Category
- Configure Auditing
- Start Auditing
- Stop Auditing
- Plot Indices
- Bulk Configuration
- Add (Delete, Modify) New Loop
- Collect Data
- Delete Dataset
- Open Dataset
- Add New Model
- Delete Model
- Identify Model
- Add New Tuning
- Delete Tuning
- Tune
- View Tuning Log
- Add New Report
- Delete Report
- Configure Report

Most of the icons are context-sensitive, i.e. are made available only when it makes sense (for example the Add New Tuning icon is active only when the Tuning function is selected on the Tree, while the Delete Tuning or Tune icons are active when a given, existing Tuning item is selected). The correspondent functions can also be activated from the Tools menu.

ABB. All rights reserved

4.5 *Configuration Tree*

The Configuration Tree is located in the left pane of the LPM Client Tool window and is the key for easy access to information and operation of the development activities of the LPM Client Tool.

The Configuration Tree contains the Tuning tab and the Auditing tab. The Tuning tab displays all the configured control loops, organized by Plant Areas. Up to 15 Areas are available in LPM. Under each Area, the loops are displayed in the same order in which they are entered. The user may select the cut and paste after or paste before options of the right click menu of a loop to change its position within the tree. For each loop there are 4 different sub-branches (Figure 38) each one able to control the following activities:

Data – for Data Collection
Model – for Model Identification
Tuning – for optimal tuning and performance evaluation
Log – for recording tuning information for each loop

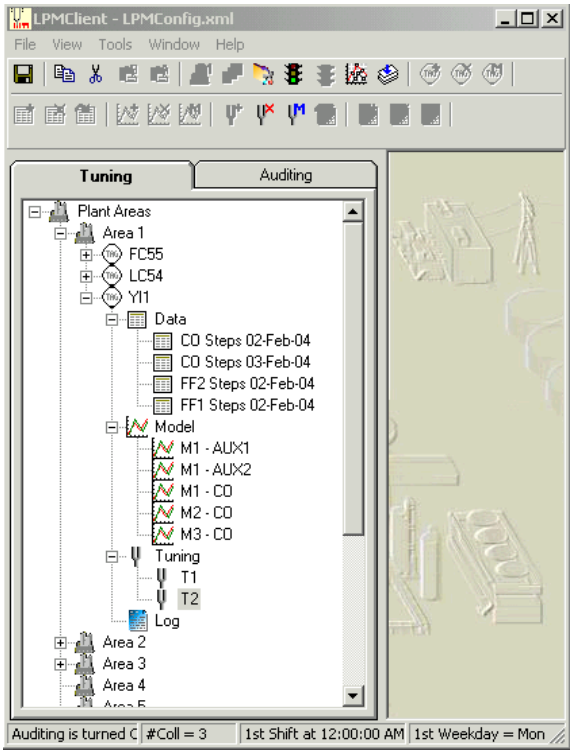


Figure 38 - Tuning Tree Structure

The highest level on the Tuning tab of the Configuration Tree consists of the 15 main branches related to Plant Areas. The following maps out these branches and the corresponding items listed under each branch item.

Area #
    Loops
        Data
            <List of Data Collection Files>
        Model

ABB. All rights reserved

<List of Identified Model Files>

Tuning

<List of Tuning Files>

Log

<List of Log Files>

The highest level on the Auditing tab of the Configuration Tree (apart from the Auditing Configuration form) consists of three main branches. The Plant Area branch shows all of the plant areas and the loops assigned to each area (same as in Loop Tuning). Each loop is also assigned to a loop category during loop configuration. There are several pre-defined loop categories (e.g. liquid flow, vapor flow, temperature) and several user-defined categories. The Loop Categories are used to define specific data collection parameters for auditing (e.g. sampling period and duration). This is accomplished by selecting Configure from the right click menu of the desired category. The third branch is for Reports. The user may configure multiple reports with custom auditing information to be automatically generated each week. Standard loop ranking reports are also automatically generated (see Figure 39 - Main Branches in the Auditing Tab).

Figure 39 - Main Branches in the Auditing Tab

The following maps out these three branches and the corresponding items listed under each branch item.

Plant Areas

- Area #
  - Loops

Loop Categories

- Liquid Flow
- Vapor Flow
- Temperature
- Composition
- Pressure
- Tight level
- Buffer Avg Level
- Other Self-regulating

ABB. All rights reserved

- Other Integrating
- User Category 1
- User Category 2
- User Category 3

Reports

Items within a branch of a tree can be made visible/invisible by expansion/contraction of the branch. If a branch on the tree has a plus symbol, this means there is a list of sub-branches (or leaves) within this branch. To see the sub branches, expand the branch by left clicking on the plus symbol (or optionally, double clicking on the branch itself). Likewise, to contract the visible sub branches on a branch of the tree, contract the branch by left clicking on the minus symbol (or optionally, double clicking on the branch itself).

The Configuration Tree has three basic functionalities:

- List specific items (e.g., datasets) as a sub list under a branch.
- Start a specific tool (i.e., bring up a window in the pane on the right side of the LPM Client Tool window that relates to the specific tree item).
- Drag and drop capability.

Details on the operation of these branches are discussed in the following sections, identified by the main activities and the Plotting Tool. Each branch on the Configuration Tree has either a green check mark or a red X warning sign associated with its icon. This is a quick indicator to the user to identify if all required information has been entered into the data entry forms.

Each data entry form will have an asterisk next to each required item. If the data entry form does not have the required information entered, a warning sign on the icon for that data entry window will appear. When all required information has been entered, the warning sign changes into a check mark.

ABB. All rights reserved

5. Tuning

5.1 *Configuration*

The first step in using LPM is related to configuration setup. Control loops can (actually should) be grouped by Plant Area to ease maintenance and trace ability.

5.1.1 Area Configuration

Each Plant Area is configured separately. In order to do so, right click on the chosen Area icon and select the Configure item as shown in Figure 40 - Area Pop-up Menu

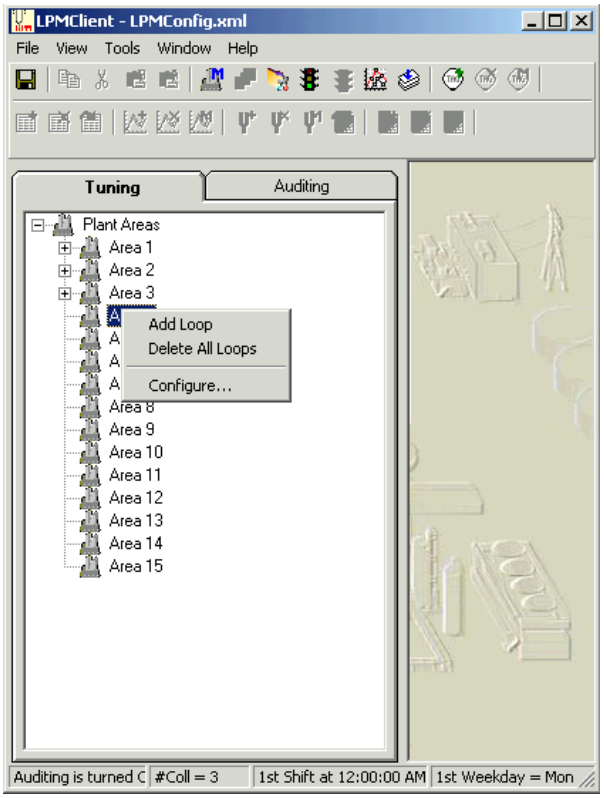


Figure 40 - Area Pop-up Menu

A dialog box similar to the one shown in Figure 41 - Area Configuration Form will allow providing the desired name to the selected Area and a short description. Furthermore the form allows specifying the Data Source Type from a drop-down menu, with all the related information. It should be noted that LPM allows three different options for Data Source Type:

- OPC DA - which is the default choice for all the DCS
- Infi90 – which is an additional option for former Bailey DCSs (Network90, Infi90, Symphony/Harmony)
- Knowledge Manager / cpmPlus History- which provides the robust data acquisition of a historian and Web based reporting

ABB. All rights reserved

Before entering into details about Data Source types it must be underlined that:

There is a single Data Source (OPC server, Infi90 or KM) for each Plant Area.

As a result, Area should be identified not only on the base of geographical partitioning of the plant, but also taking into consideration where the actual data servers are placed.

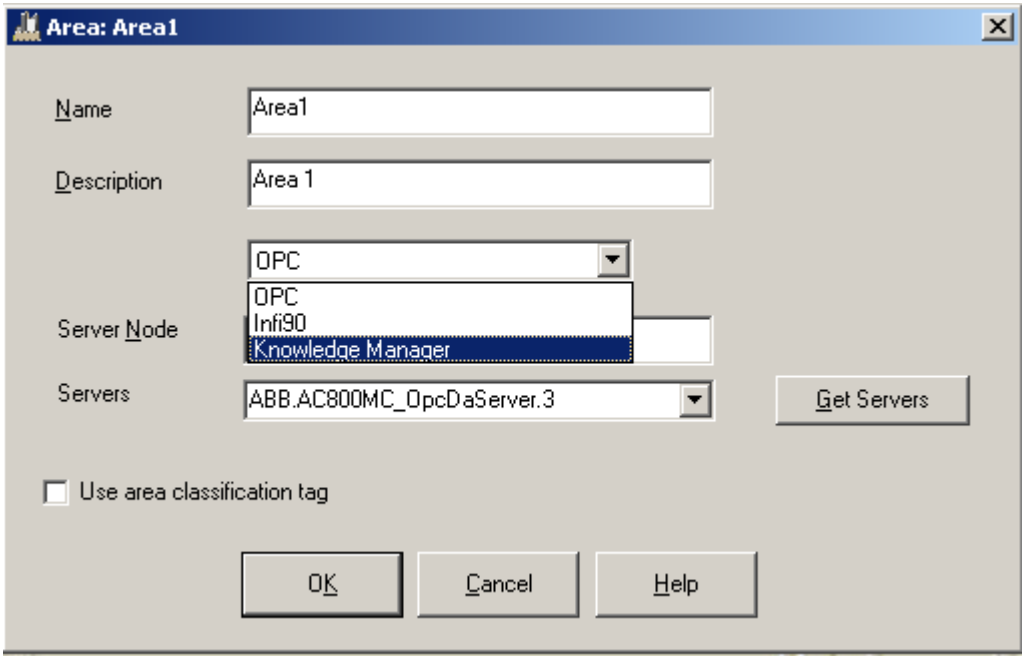


Figure 41 - Area Configuration Form

5.1.1.1 OPC Data Source Type Configuration

If the OPC option is selected the user will have to act through the form shown in Figure 41 - Area Configuration Form entering the Server Node name and specifying which server must be used for gathering information. Through the Get Servers button it is possible to let the software automatically list the names and addresses of the OPC servers installed on the machine specified in the Server Node textbox.

5.1.1.2 Infi90 Data Source Type Configuration

If the Infi90 option is selected the form will dynamically change as shown in Figure 42 - Area Configuration Form for Infi90 Connection. In order to properly configure the communication the user will act on the Setup button. By clicking on it the configuration form shown in Figure 433 will made available.

Attention: ***For an Infi90 data connection, the Source Addresses defined in the loop configuration window should be entered in the following format: Infi90Loop/PCU/Module/Block***

ABB. All rights reserved

User could specify the Communication Loop Type (i.e. Serial Port Module, Plant loop, InfiNet, etc.), the Serial Port Number and the Baud Rate.

In the lower box the user may check the status of CIU/CPM settings. Please refer to the proper Infi90/Harmony User Manual for more details.

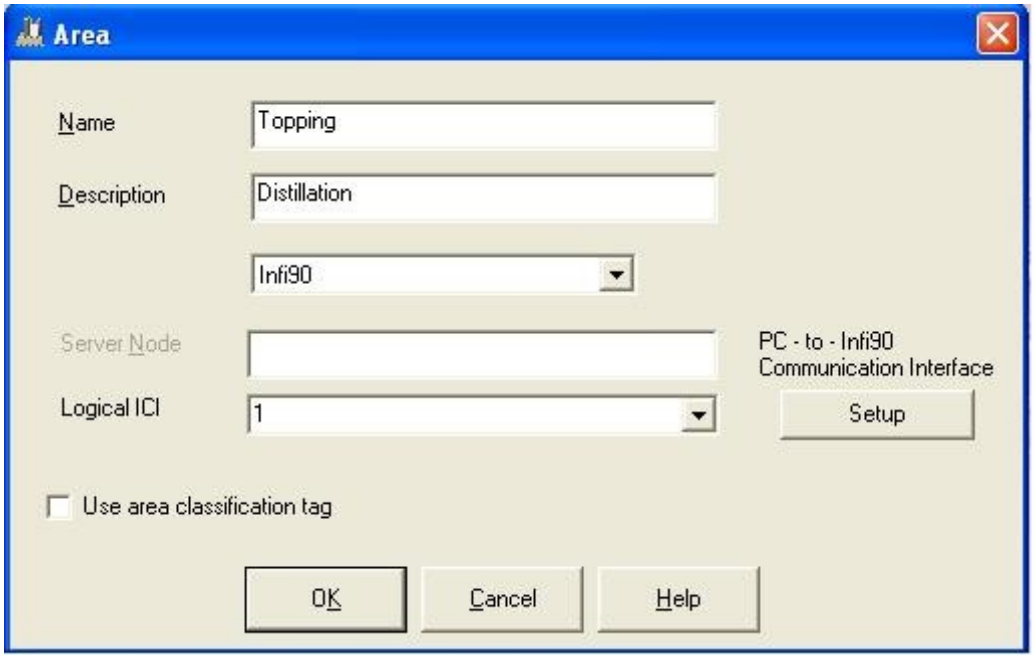


Figure 42 - Area Configuration Form for Infi90 Connection

ABB. All rights reserved

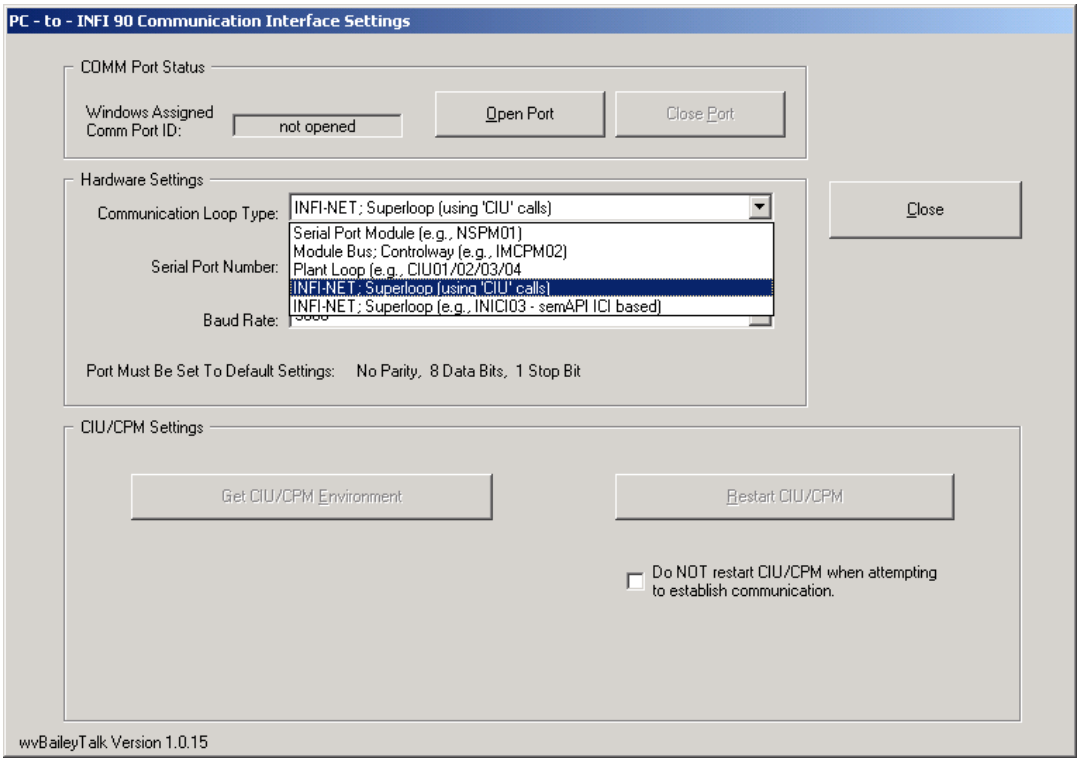

PC - to - INFI 90 Communication Interface Settings
COMM Port Status
Windows Assigned Comm Port ID:
not opened
Open Port
Close Port
Hardware Settings
Communication Loop Type:
INFI-NET; Superloop (using 'CIU' calls)
Serial Port Module (e.g., NSPM01)
Module Bus; Controlway (e.g., IMCPM02)
Plant Loop (e.g., CIU01/02/03/04
INFI-NET; Superloop (using 'CIU' calls)
INFI-NET; Superloop (e.g., INICI03 - semAPI ICI based)
Serial Port Number:
Baud Rate:
Port Must Be Set To Default Settings: No Parity, 8 Data Bits, 1 Stop Bit
Close
CIU/CPM Settings
Get CIU/CPM Environment
Restart CIU/CPM
Do NOT restart CIU/CPM when attempting to establish communication.
wvBaileyTalk Version 1.0.15

Figure 43 – Communication Interface Setting

Note : ***If an INFINET SemAPI ICI based is used as Infi90 communication interface it is strictly recommended to use a dedicated ICI connection for the LPM Infi90 communication. The simultaneous presence of LPM and additional external applications or Information Systems (e.g. Composer) sharing the same communication interface for the connection to Infi90 generates conflict in the data communication and prevent the correct data acquiring between LPM and the Infi90 Data source.***

5.1.1.3 Knowledge Manager 6.1 / cpmPlus History

Details regarding this connectivity mode are explained in Configuring the KM Server 7.3.

5.1.2 Loop Configuration

In order to configure all the loops of interest in the area, the user has two options:

- Enter manually all the requested loop information.
- Bulk Configuration Utility to automatically import from an external file (i.e. an MS Excel file) all the needed information.

ABB. All rights reserved

5.1.2.1 Manual Entry

As shown in Figure 444 in order to manually enter the requested information, right click on the Area icon, and select the Add Item option to add each loop. An alternative option it is possible to add a new loop using the Menu button in the top of the window.

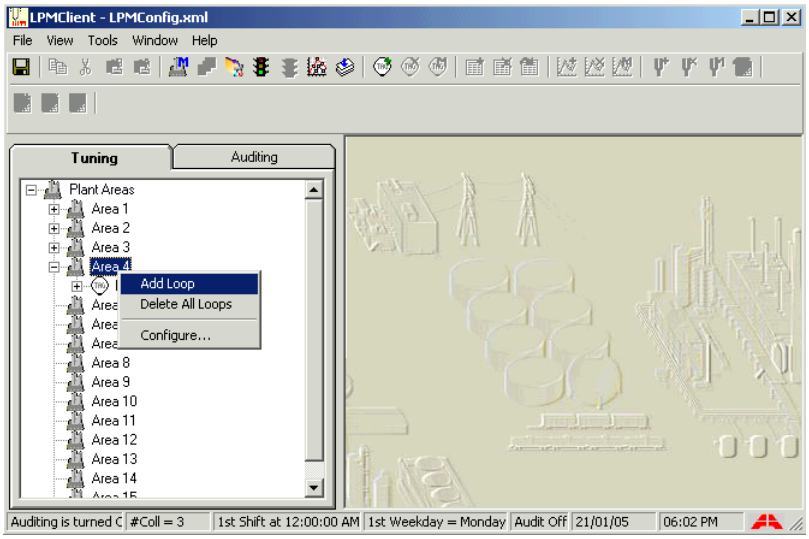


Figure 44 - Add Loop

A dialog box similar to the one in Figure 45 - Add Loop Dialog, will allow you to insert the name of the new loop.

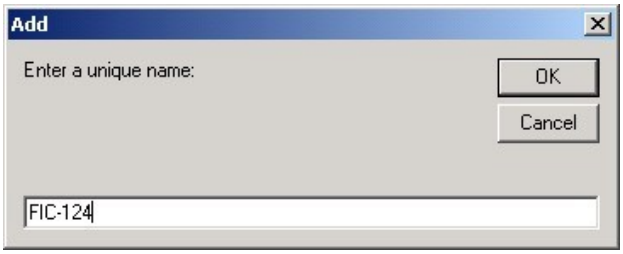


Figure 45 - Add Loop Dialog

After clicking on the OK button, the Tree will change, and a structure similar to the one shown in Figure 46 - Loop Tree View Structure will appear.

ABB. All rights reserved

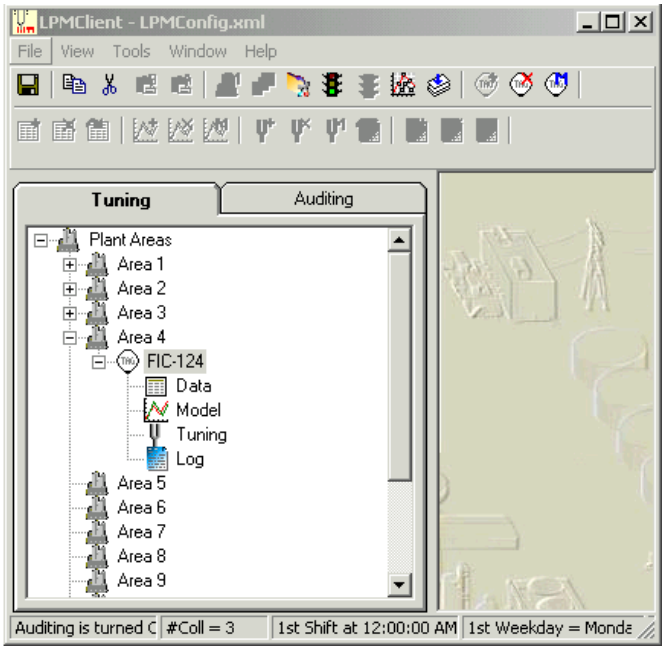


Figure 46 - Loop Tree View Structure

By right-clicking on the Loop name of the Plant Tree it is possible to access the Loop Configuration Form, shown in Figure 47 - Loop Configuration Form. The same result can be obtained by pressing the relevant menu button in the top of the LPM window.
Taking into consideration Figure 47 - Loop Configuration Form, the top section of the window "**Loop Configuration**" is devoted to general information for the Loop (name, description), which may be edited.
Two drop-down menus allow selecting the Loop category (Figure 48 - Loop Category drop-down Menu) and the PID Controller Type (Figure 49 - PID Controller Type drop-down Menu).

On the top right size of the form, three check-marks are available:

- Audit active allows to specify if the selected loop should be included in the Auditing data collections;
- Report active allows to set if the selected loop auditing figures should be included in the periodical reports.
- Cascade Master (Outer) specifies that the selected loop is an external loop in a cascade configuration. See chapter 5.6 on Cascade loop for a description on the procedure to configure and tune Cascade Loop.

ABB. All rights reserved

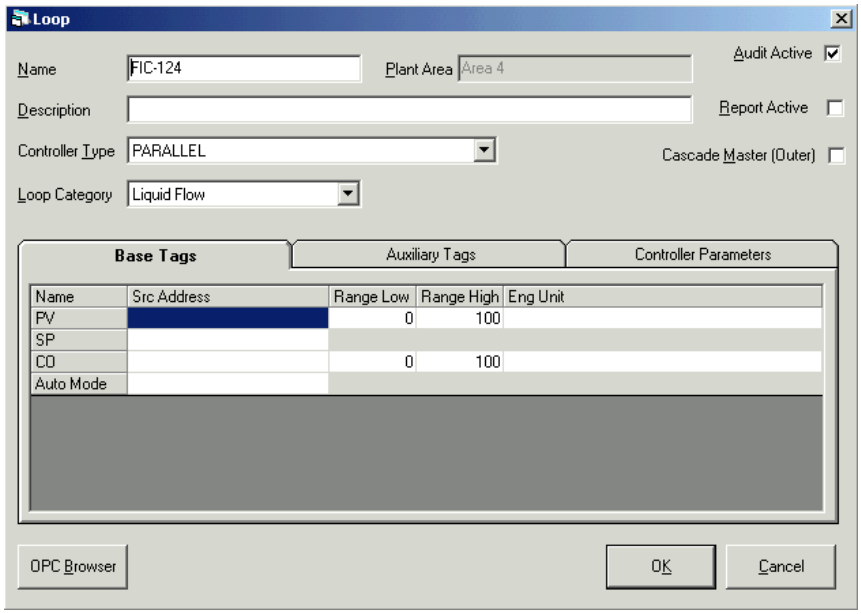


Figure 47 - Loop Configuration Form

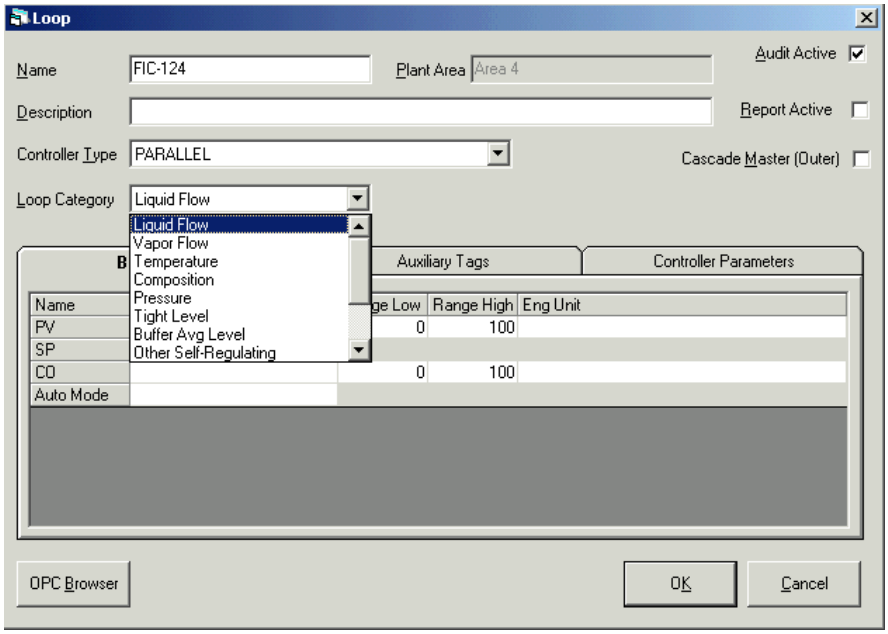


Figure 48 - Loop Category drop-down Menu

ABB. All rights reserved

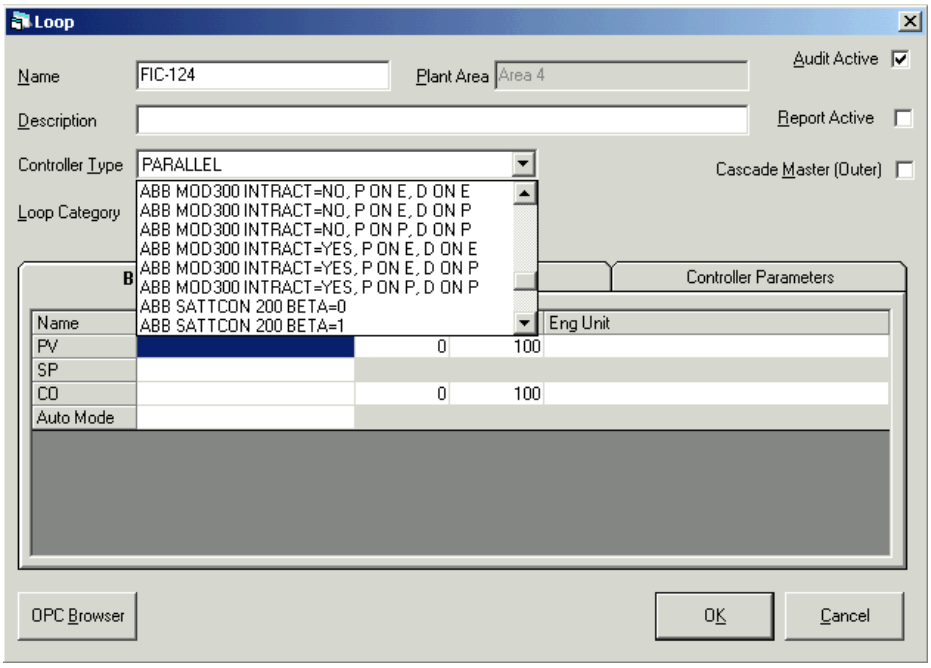


Figure 49 - PID Controller Type drop-down Menu

The bottom part of the form is composed of three tabs:

- Base Tags
- Auxiliary Tags
- Controller Constants

The Base Tags tab, shown in Figure 47 - Loop Configuration Form allows editing of the basic information (ranges, engineering units, OPC tagnames) for the loop PV, SP, CO and Auto Mode tags.

<u>Attention</u>: ***Most PID controllers also include some form of scaling to normalize the equations and the resulting gain parameters. Many controllers use the PV range and CO range for scaling, while some controllers include a separate gain scaling parameter. In order for LPM to obtain proper tuning parameters for industrial PID types, it must obtain the appropriate PID loop range values or gain scaling parameters. For many PID types, LPM defines the scaling by applying the PV range and CO range defined in the LPM Loop Configuration form.***
For PID types that use a separate scaling parameter (e.g. Bailey), LPM defines the scaling by applying the Gain Scaling Parameter defined in the LPM Loop Configuration form (see Figure 50 - Gain Multiplier for Bailey PIDs). Still referring to the Bailey INFI90 case (or for other PID types with similar scaling features), a special care must be spent in case of Cascade configuration for the Outer Loop. In this case the gain scaling parameter should be defined as the product of the actual PID Gain Multiplier (K) and the factor which actually rescales the outer PID CO to the Inner Loop SP[♣]. Please note that this correction must be done manually for both the possible configuration options (Manual Entry or Bulk Data Configuration Utility).

[♣] *If the Inner Loop SP ranges between 0 and 20000, the Outer PID CO (which is obviously limited between 0 and 100) must be multiplied by a factor 200. Therefore, in this case, the Outer PID Gain Multiplier will have to be multiplied by a factor 200.*

ABB. All rights reserved

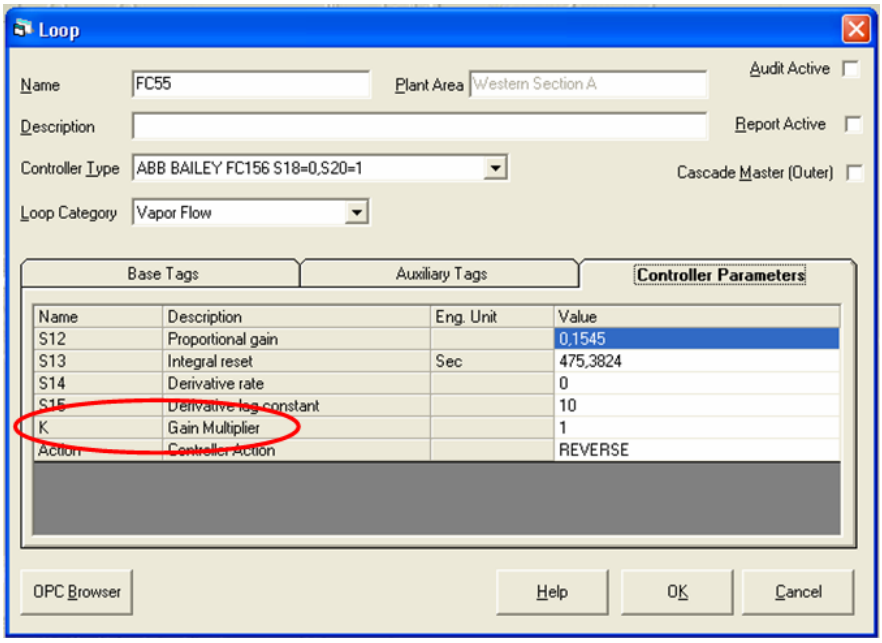


Figure 50 - Gain Multiplier for Bailey PIDs

The Auxiliary Tag tab is devoted to insert the needed information for up to 5 auxiliary variables affecting the loop as shown in Figure 51 - Auxiliary Tags Tab. In particular, each auxiliary variable can be configured as a Feedforward variable by checking the FF box on the form.

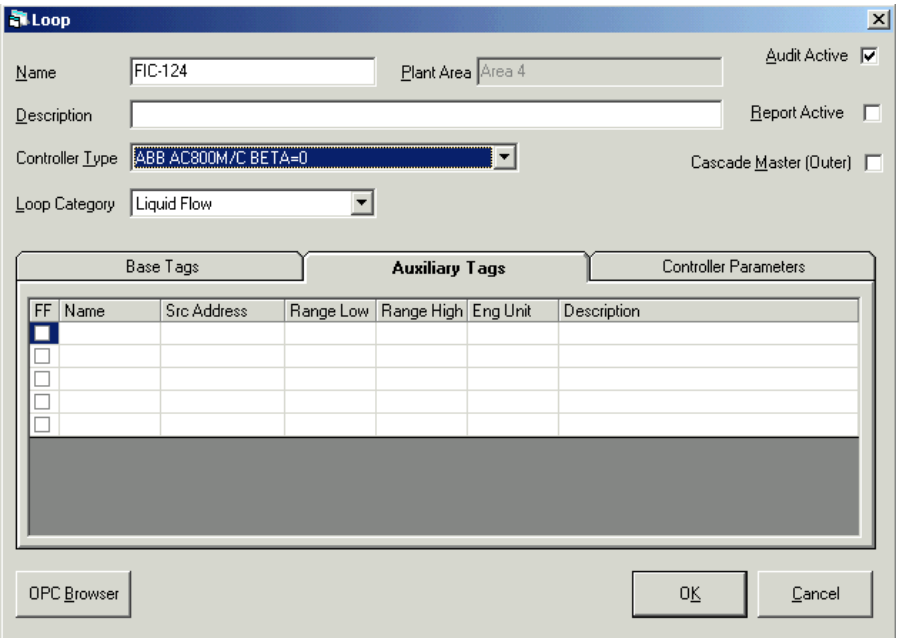


Figure 51 - Auxiliary Tags Tab

ABB. All rights reserved

Finally the Controller Constants tab is devoted to introduce of controller constants. These may be entered for the existing tuning or left empty. After each tuning activity, which will be saved in the Logs (see § 6.5), the Controller Constants will be updated also in this form. Please note that this form is context sensitive, i.e. the list changes according to the PID controller selected. Figure 52 - Controller Constants Tab shows as an example the parameters of the ABB Bailey Function Code 156 PID controller.

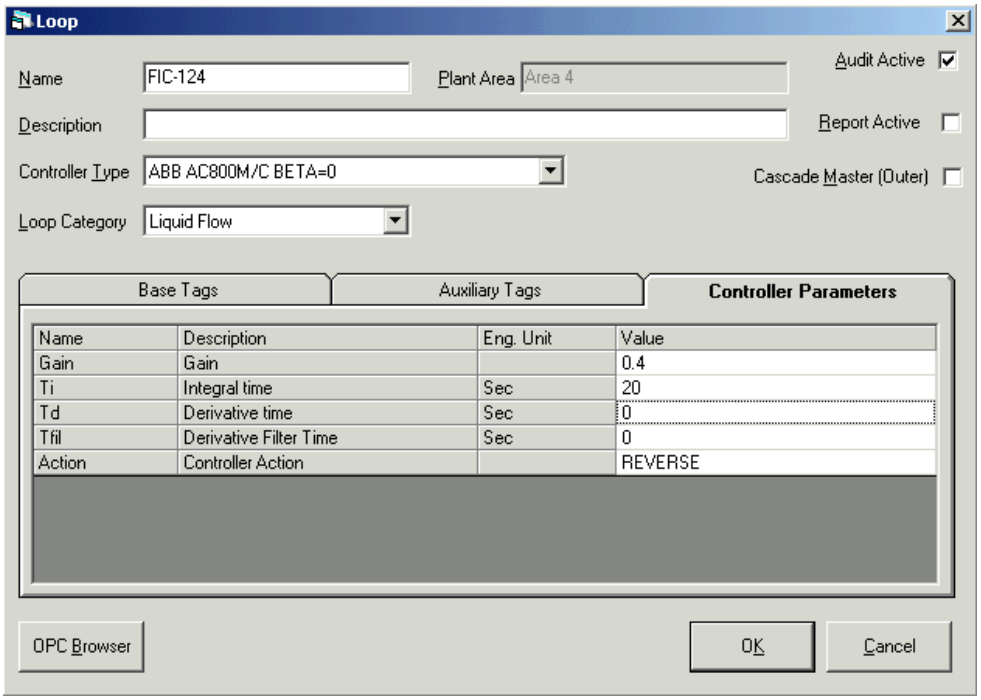


Figure 52 - Controller Constants Tab

5.1.2.2 Bulk Data Configuration Utility

As an alternative to the Manual entry mode, all the needed loop data for a given project may be entered into LPM, through the Bulk Configuration utility. To simplify the creation and configuration of many control loops in LPM, the user is able to import this information from a Microsoft Excel Workbook.

The Bulk Data Configuration Utility tool could be launched from the devoted icon on the toolbar or from an item in the Tools menu, as shown below in Figure 53 - Launching the Bulk Data Configuration Utility.

ABB. All rights reserved

Figure 53 - Launching the Bulk Data Configuration Utility

By selecting this option a form will be shown as in Figure 54 -The Bulk Data Configuration Utility Window.

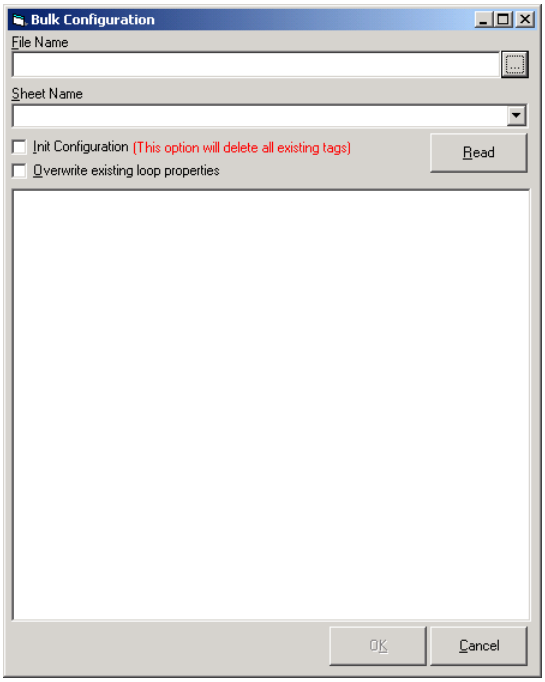


Figure 54 -The Bulk Data Configuration Utility Window

Clicking on the browse button (…) at the right of the 'File Name' field, the user will be prompted to browse the PC to select the proper Microsoft Excel workbook to be imported (see Figure 55 - The File Selection Dialog).

ABB. All rights reserved

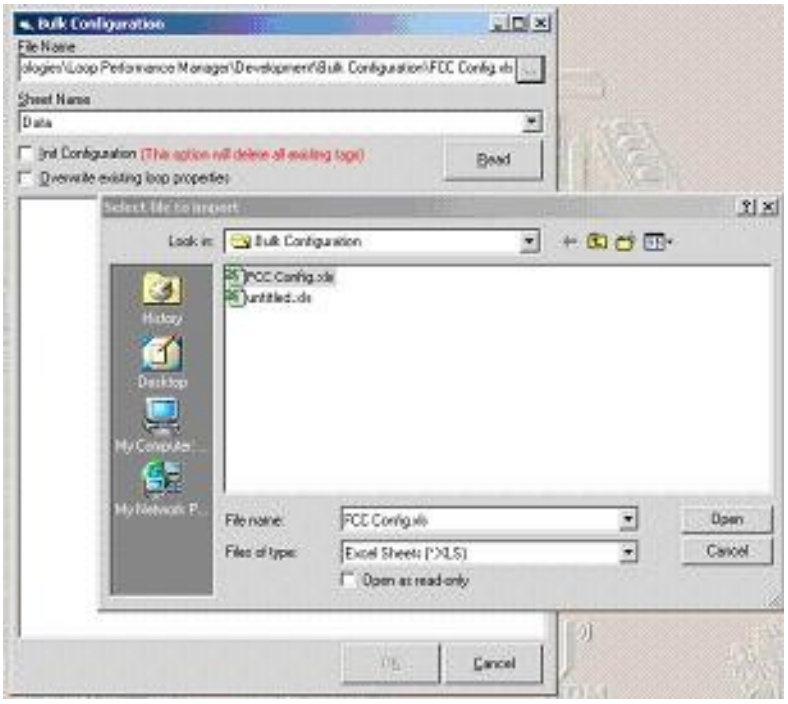


Figure 55 - The File Selection Dialog

After selecting the file (and selecting the desired sheet from the 'Sheet Name' list), the user can check options to either 'Init Configuration' or to 'Overwrite existing loop properties'.

- If the "Init Configuration" box is checked all the existing tags will be deleted before importing the new tag configurations;
- If the "Overwrite existing loop properties" box is checked the properties from Excel file will overwrite the existing loop properties in case of redundancy.

Afterwards the user can click on the "Read" button: this will make the utility access and read the Excel spreadsheet. A list of warnings and/or errors is given in the working area.

Finally the user can click on the "OK" button: this will activate the utility, which will create the tags and configure them with the data found in the Excel spreadsheet.

ABB. All rights reserved

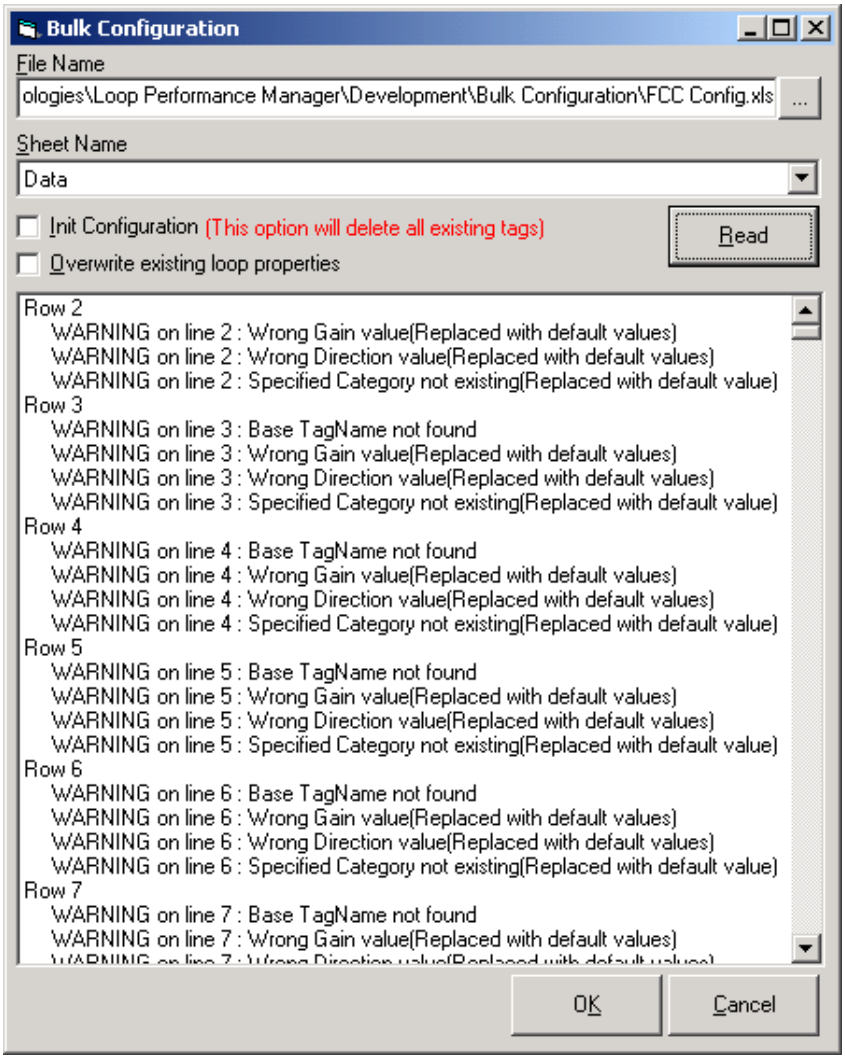

Bulk Configuration
File Name
ologies\Loop Performance Manager\Development\Bulk Configuration\FCC Config.xls
...
Sheet Name
Data
Init Configuration (This option will delete all existing tags)
Overwrite existing loop properties
Read
Row 2
WARNING on line 2 : Wrong Gain value(Replaced with default values)
WARNING on line 2 : Wrong Direction value(Replaced with default values)
WARNING on line 2 : Specified Category not existing(Replaced with default value)
Row 3
WARNING on line 3 : Base TagName not found
WARNING on line 3 : Wrong Gain value(Replaced with default values)
WARNING on line 3 : Wrong Direction value(Replaced with default values)
WARNING on line 3 : Specified Category not existing(Replaced with default value)
Row 4
WARNING on line 4 : Base TagName not found
WARNING on line 4 : Wrong Gain value(Replaced with default values)
WARNING on line 4 : Wrong Direction value(Replaced with default values)
WARNING on line 4 : Specified Category not existing(Replaced with default value)
Row 5
WARNING on line 5 : Base TagName not found
WARNING on line 5 : Wrong Gain value(Replaced with default values)
WARNING on line 5 : Wrong Direction value(Replaced with default values)
WARNING on line 5 : Specified Category not existing(Replaced with default value)
Row 6
WARNING on line 6 : Base TagName not found
WARNING on line 6 : Wrong Gain value(Replaced with default values)
WARNING on line 6 : Wrong Direction value(Replaced with default values)
WARNING on line 6 : Specified Category not existing(Replaced with default value)
Row 7
WARNING on line 7 : Base TagName not found
WARNING on line 7 : Wrong Gain value(Replaced with default values)
OK
Cancel

Figure 56 - Warning Messages During Tag Configuration Build-Up

The above actions will populate the configuration as shown in the Figure 56 - Warning Messages During Tag Configuration Build-Up below.

ABB. All rights reserved

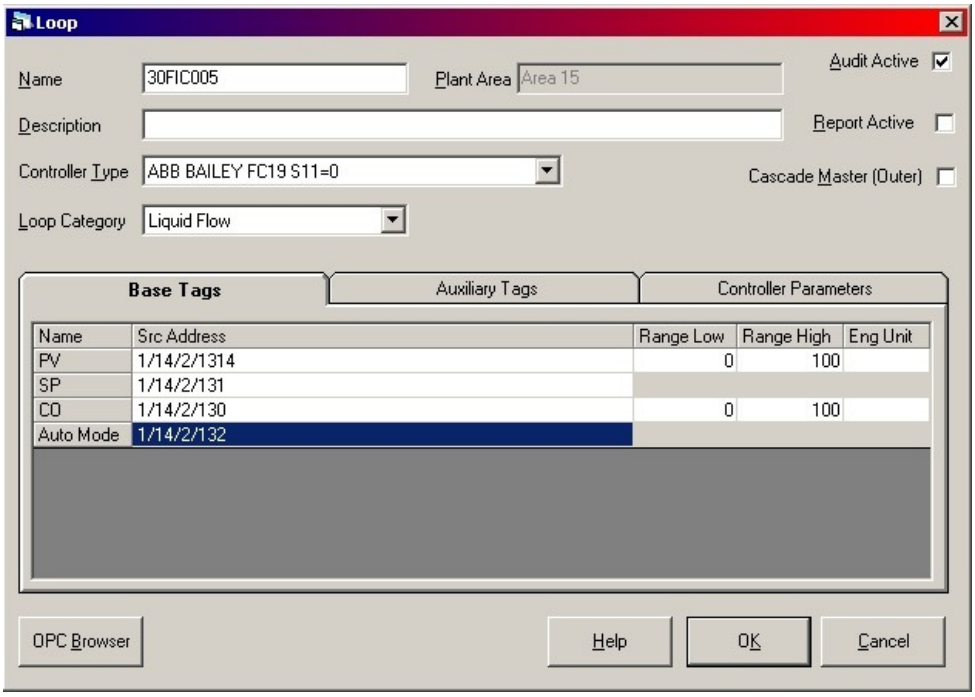


Figure 57 - Bulk Data Configuration Result

5.1.2.3 Excel File Required Format

In order to be compatible with the bulk configuration utility the Excel spreadsheet including the configuration data, should confirm with the following requirements:

Each row in the spreadsheet will represent a loop. The first row will be used for comments. The loop data will start from second row and will continue without any gap.

There will be 17 columns in the following order:

1. **Area Name** – Name of the plant area loop belongs to. Should be one of the Areas configured in LPM.
2. **Loop Name**
3. **Base Tagname** – a prefix to be used with PV Addr, SP Addr, CO Addr, and MODE Addr.
4. **PV Addr** – Address of the Process Variable. Depends on the connection type e.g. OPC, semAPI etc.
5. **SP Addr** – Address of the Setpoint.
6. **CO Addr** – Address of the Controller Output.
7. **MODE Addr** – Address of the Controller Mode e.g. AUTO/MAN.
8. Description
9. **PV Low** – Low range of PV, PV Low should be less than PV High
10. **PV High** – High range of PV
11. **CO Low** – Low range of CO, CO Low should be less than CO High

ABB. All rights reserved

12. **CO High** – High range of CO
13. **Gain** – Bailey Gain multiplier, should be non-zero positive number
14. **Beta** – PIDCONA setpoint weight for the proportional part, should be between 0 and 1 inclusive
15. **Category** – The category to which loop belongs. Should be one of the defined loop categories in LPM (see section 7.2.3).
16. **ControllerType**. Should be one of the PID controller types supported by LPM (see Appendix E).
17. **Direction**: (1) for Reverse and (–1) for Direct.

Note that the Base Tagname entry is used as a prefix in determining the complete names for the PV Addr, SP Addr, CO Addr, and MODE Addr, e.g. if Base Tagname cell = "TIC100" and PV Addr cell = ".PV/SIG", then the actual PV Addr for this loop is "TIC100.PV/SIG".

ABB. All rights reserved

5.2 *Data Collection*

After having configured the loop(s) in consideration, data may be collected to use for the Model Identification/Tuning procedure.

To collect data, right click on the "Data" icon below the desired loop name, as in Figure 58 - Data Pop-up Menu, or press the relevant menu button in the top of the LPM window.

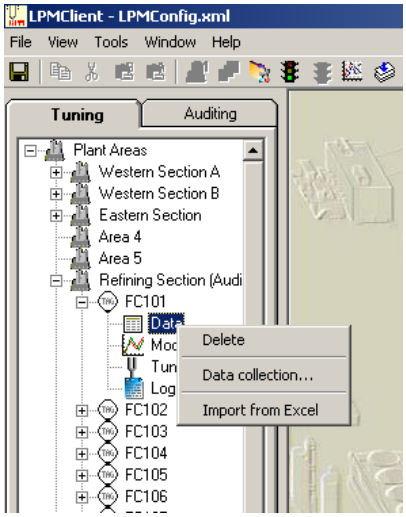


Figure 58 - Data Pop-up Menu

This will open a form similar to the one shown in Figure 59 - Data Collection Window. The user should provide the following basic information required by the Data Collection activity:

- **Sampling Rate**, i.e. the rate at which data is collected and stored for each configured tag of this loop (to be expressed in seconds);
- **Duration**, i.e. length of time for the data collection (to be expressed in minutes); Note that a data collection can always be stopped before completing the specified duration.
- **Description**, i.e. a short description that can be useful to identify the data collection at a later stage (e.g. date and time, particular plant conditions, etc.).

ABB. All rights reserved

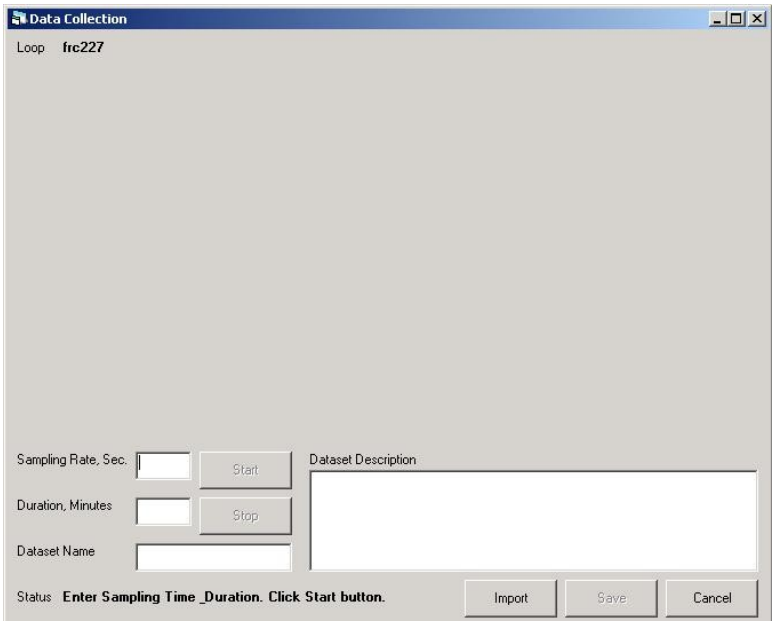


Figure 59 - Data Collection Window

Click the Start button to launch the data collection. LPM acquires via OPC: PV, CO and Setpoint and the actual signals are visualized in a graph.
During data collection, it is necessary to perform some step using the DCS Console of the Base Automation, in order to excite process and capture the loop dynamic characteristics. Step number, amplitude and period should be evaluated case by case in order to collect information reach enough.
LPM allows to perform data acquisition both with the investigated loop both in manual and close loop mode.

Figure 60 - Data Collection Example shows an example of this. The plot shows the CO (in blue) and the PV (in green).

ABB. All rights reserved

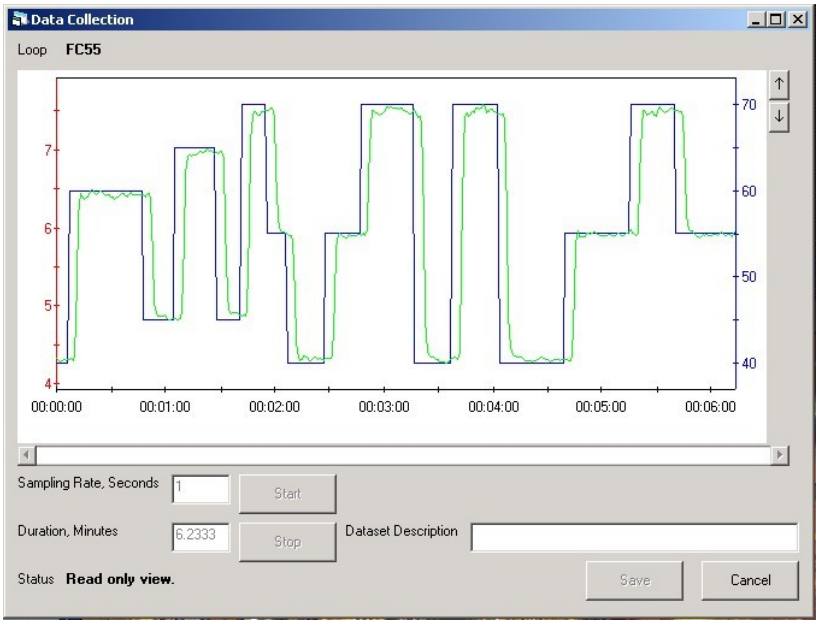


Figure 60 - Data Collection Example

When the data collected are deemed to be acceptable for modeling the user can use the dataset to build process models through the Modeling form.

5.2.1 Importing Data from MS-Excel® files

As an alternative to real-time data collection, LPM makes available to the user the possibility to import data from Microsoft-Excel® files.

Attention: ***Using imported data requires some special care about data quality. If data have been collected through Plant Historians (which represent probably the most common data repository available in process plants), the user has to check that the sampling time is adequate in order to properly catch loop dynamic behavior. A further check should validate that the Historian has applied no data compression algorithm on the data in order to improve storage efficiency. As a matter of fact, data compression can hide crucial dynamical details biasing or even preventing any effective model identification effort.***

In order to import Excel files the user must right-click on the "Data" icon below the desired loop name, as in Figure 58 - Data Pop-up Menu, or press the relevant menu button in the top of the LPM window.
This will open a dialogue box similar to the one shown in Figure 61 – Excel Import: Dataset Name Selection

ABB. All rights reserved

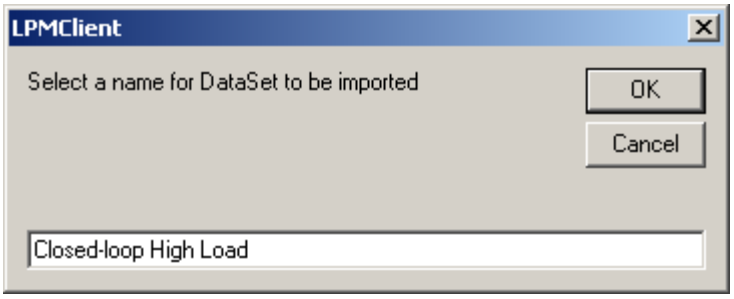


Figure 61 – Excel Import: Dataset Name Selection

After having inserted the Data Set name and clicked on the "OK" button, a special form (Figure 62 – Excel Data Import Form) will appear to help specifying the import information.

The top section requires the user to insert the name of the Excel file containing the related data and which of the possible worksheets must be used.

The second section allows to specify which is the column containing the TimeStamp and the time format used in the excel file. The third and last section allows to specify how many initial rows are devoted to comments or other indications and must be skipped when importing data and to identify which is the column containing SP, CO and PV data, together with a basic strategy to deal with possible bad quality data in the file.

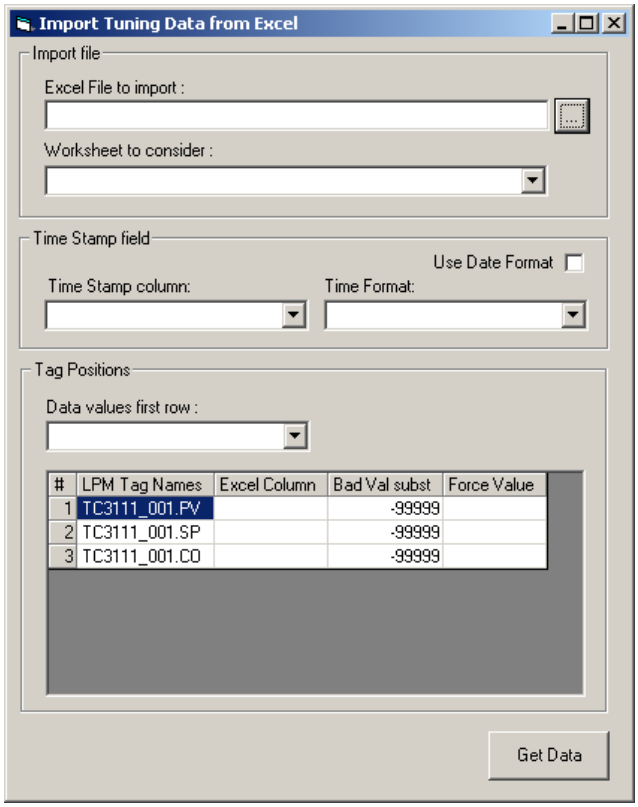


Figure 62 – Excel Data Import Form

ABB. All rights reserved

After having inserted all the proper information in the above described fields, the user must click on the “Get Data” button. This will trigger the actual data import: at the end the new data set will appear in the tree as any other dataset, as shown in Figure 63 – Excel Imported Dataset.

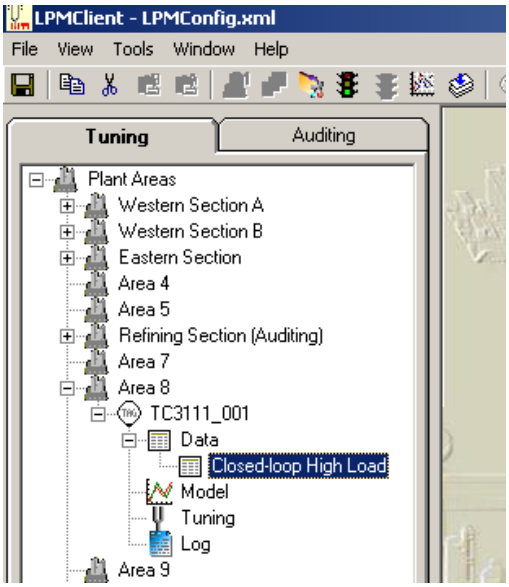


Figure 63 – Excel Imported Dataset

When displayed the imported dataset will automatically report in the section the information about the Excel file it has been imported from (Figure 64 – Displaying Excel-imported Datasets).

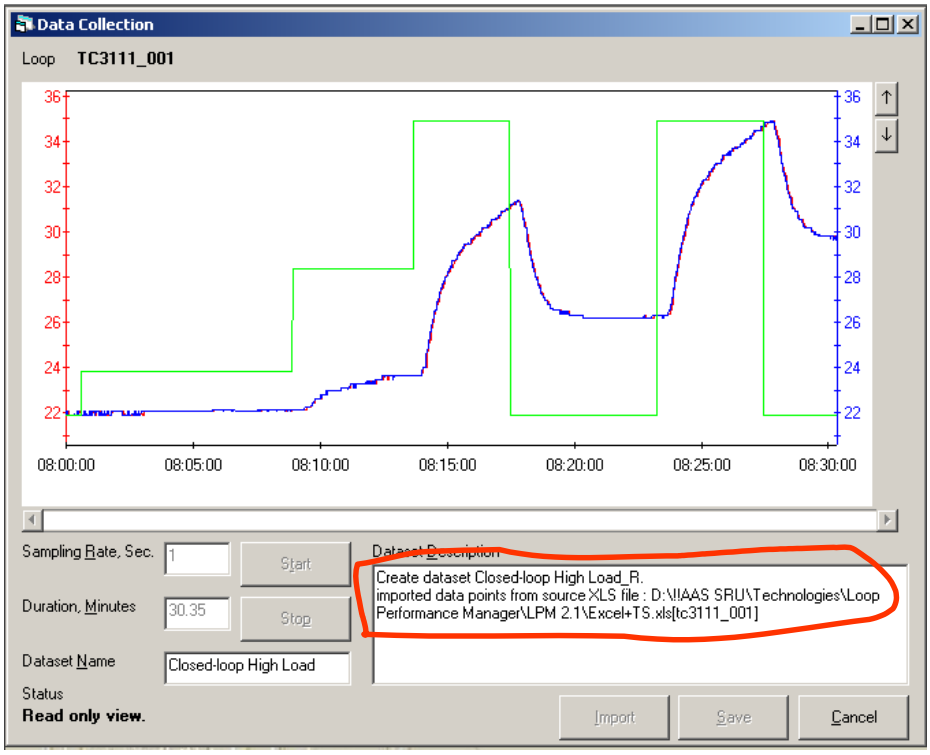


Figure 64 – Displaying Excel-imported Datasets

ABB. All rights reserved

In summary, in order to be compatible with the import utility, the Excel spreadsheet must fulfill the following requirements:

Each row in the spreadsheet must represent data at a given time. A number of rows at the top of the spreadsheet can be safely used for comments and will be removed during the import procedure that will start from what identified as “Data Values First Row”. The loop data will start from “Data Values First Row” and must continue without any gap.

For each loop the file must include 4 columns in any order:

1. **Timestamp** – uniquely identifying the time when the sample has been collected
2. **SP** – the value of the Setpoint at the related time
3. **OP -** the value of the Control Output at the related time
4. **PV** – the value of the Process Variable at the related time.

ABB. All rights reserved

5.3 *Modeling*

A model of the process loop being controlled is required to enable tuning of the associated controller. The modeling tool seeks to obtain a mathematical model describing the dynamic behavior of the loop. It is done by establishing a model structure and determining its parameter values based on data logged from the process. For the model identification to be successful there must be suitable excitation in the control output and process variable during the logging period. That is the only way in which the required information about the process behavior can be concluded from the logged data.
The model type (structure) can be selected manually or be determined automatically. The model obtained is presented numerically by its parameters and graphically by its step and frequency responses. Validation of a model is normally performed using a different data set than that used for model fitting.
In addition to modeling by identification, it is also possible to specify a model manually, simply by selecting a structure and manually entering the parameters. The model type and parameter values are defined for continuous time models.

The model structure supports various continuous time model types, from simple integration with gain, to third-order models with integration, one zero, gain and deadtime.
For those cases when the model type is unknown, LPM will determine the type automatically.
A suitable procedure for modeling is suggested by the following sequence of subheadings, corresponding to the tabs on the Model Identification form.

To add a new model definition for any of the three model definitions branches, right click on Model in the Tree and select **Add Item**. This will open a dialog box similar to the one shown in Figure 65 - Add Model Form.

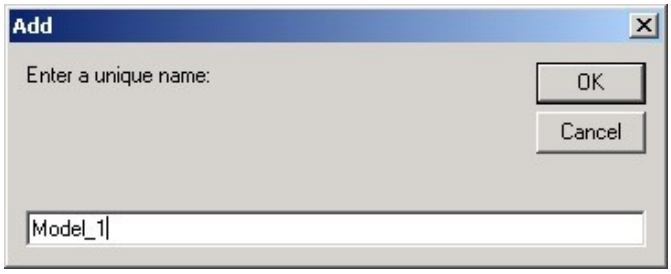


Figure 65 - Add Model Form

ABB. All rights reserved

A model definition includes all of the relevant information about the model and how it was created (e.g. structure, parameters, datasets, filter settings, etc.).

Operations with a specific model definition consist of (see Figure 66 - Model Operations):

- Delete/Copy/Cut/Paste: Right click specific model definition and select appropriate item.
- Model Identification: Right click specific model and select Identify (or optionally double click specific model definition).
- By using Menu Buttons it is possible to “Add a new Model”, “Delete a Model” and “Identify Model”

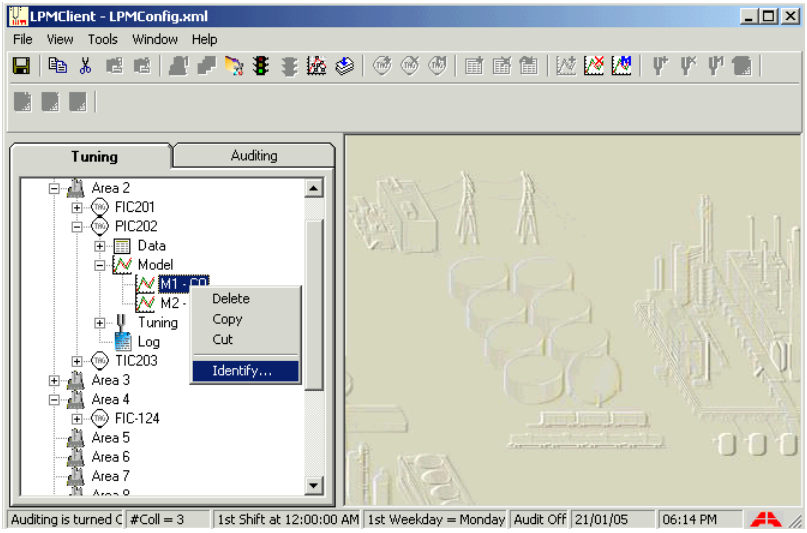


Figure 66 - Model Operations

This will open the main Model Identification form in Figure 67 - Model Identification Screen.

ABB. All rights reserved

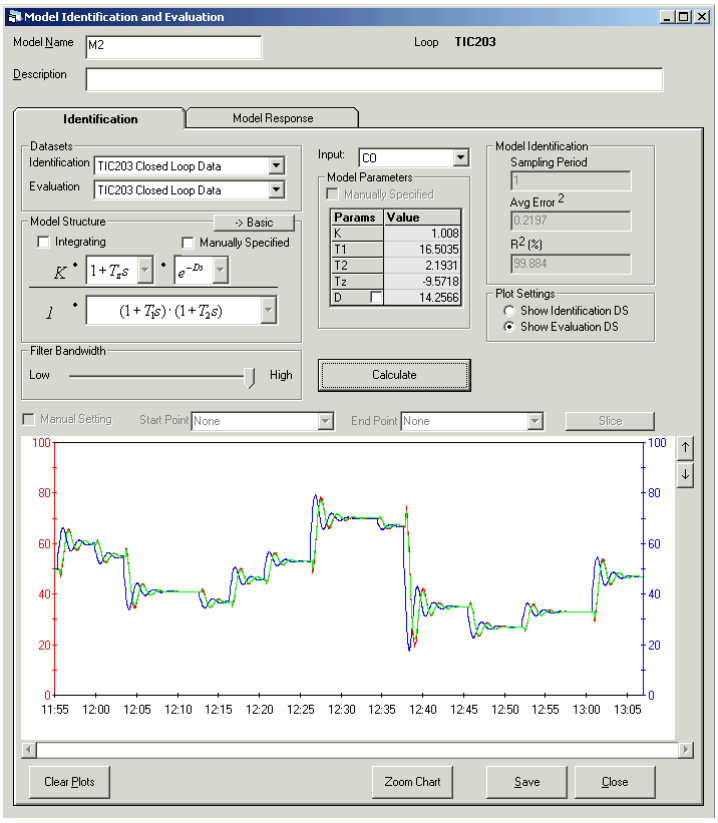


Figure 67 - Model Identification Screen

At the top of the window, the user may enter the Model Name and a short description. On the right the name of the selected loop is automatically displayed.
If you are opening an existing LPM model, the model window will show the calculated parameters and charts.

The Model Identification screen is composed of two tabs:

- The **Identification** Tab used to create a process model
- The **Model Response** Tab used to evaluate the dynamic behavior of the created model.

5.3.1 Identification Tab

The Identification Tab is used to create a model by identifying or specifying model parameters.

5.3.1.1 Collected Data elaboration

On the top-left corner a frame labeled "*Datasets*" includes two drop-down menus that allow the user to select a dataset to be used for Identification and one for Evaluation, from the datasets listed under the Data branch. In the example below (Figure 68 - Model Identification Example) the identification dataset is "*YI1 CO Steps 02-Feb-04*", while the evaluation one is "*YI1 CO Steps 03-Feb-04*".

ABB. All rights reserved

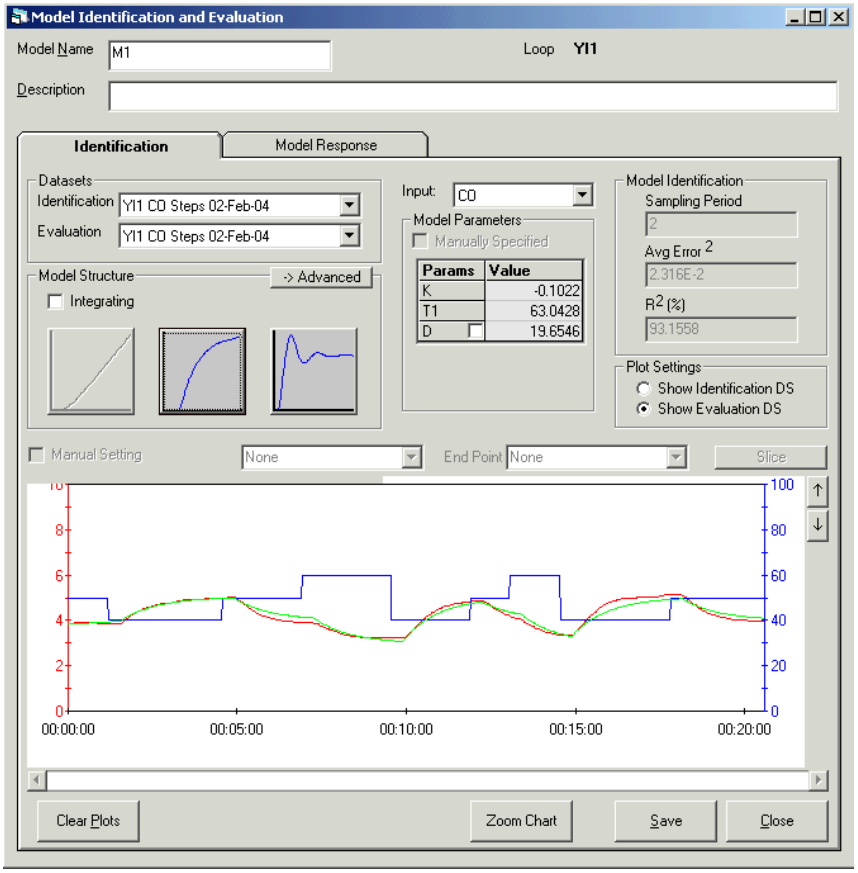


Figure 68 - Model Identification Example

After having selected the datasets, the user should specify the independent variable (*Input*) in the model identification procedure. There are mainly 3 possibilities:

- If the loop has been configured without any auxiliary variable (i.e. without any Feedforward variable) and it is not a Cascade Outer Loop, the only choice is the loop Control Output (CO), and it will be pre-selected (see Figure 68 - Model Identification Example).
- If the Loop is configured with Feedforward variables, it is possible to use these external variables as Independent Input. See the example below (Figure 69 - Model Identification Example 2), where the loop Y1 has been configured with two possible FF variables (named AUX1 and AUX 2) and then the drop-down menu shows all possible input variables. Note that the user must check the FF checkbox in the configuration to allow the auxiliary signal to appear as a Feedforward variable in this form.
- If the loop is an external Loop of a Cascade configuration, it is possible to select the Internal PV as Independent variable. See chapter 5.6 on Cascade loop for more details.

ABB. All rights reserved

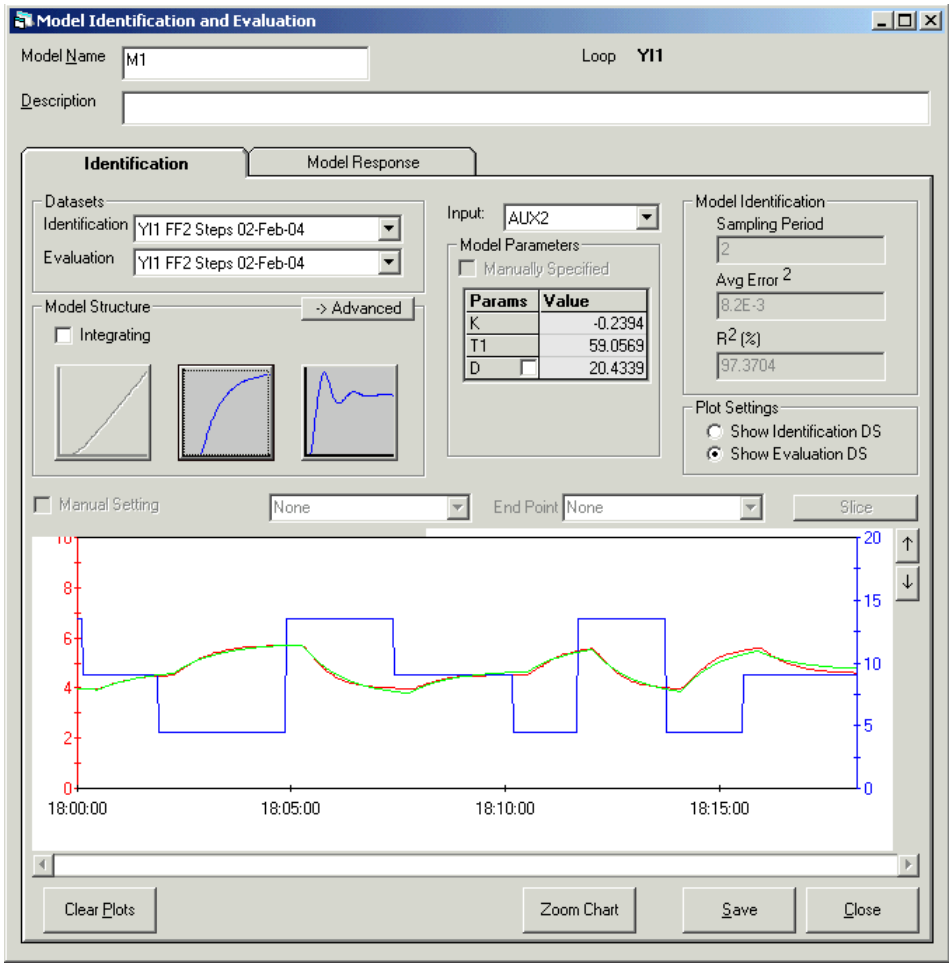


Figure 69 - Model Identification Example 2

The signals used for identification are displayed in the bottom part of the **Identification** Tab. LPM offers the possibility to select only a slice of the total acquired data. This is performed by choosing the "*Manual Setting*" check box; afterward the requested slice is defined by means of the two combo boxes: "*start point*" and "*end point*".

As shown in Figure 70 - Model Identification Slice selection the selected piece of data are highlighted in light blue. In case the defined data is not correct, it is possible to go back to the complete data set by means of the push button "*DelSlice*".

ABB. All rights reserved

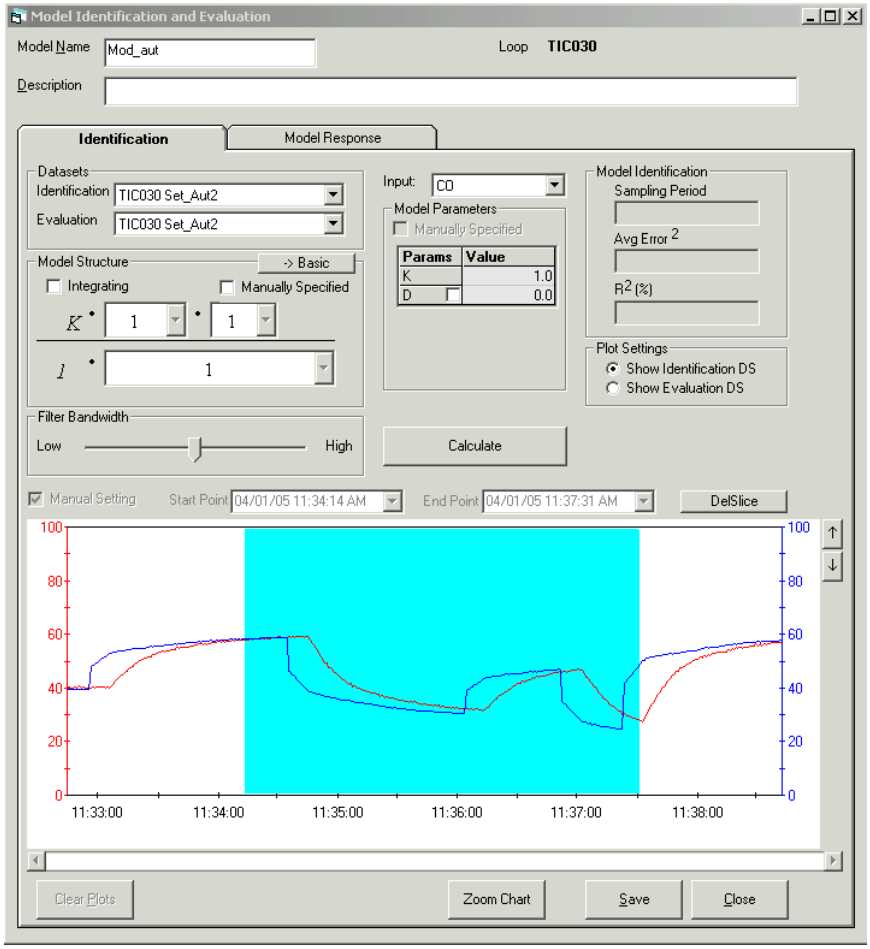


Figure 70 - Model Identification Slice selection

Below the "*Datasets*" frame is the "*Model Structure*" frame, which allows the user to specify the structure of the model to be identified. LPM has two "working modes" to define the Model structure, designed to aim to different Plant Engineers:

- Basic Mode – for Engineers with low experience in Control Theory
- Advanced Mode – for Engineers with high experience in Control Theory

The selection between the two modes is carried out by means of the button in the top-right of the form "*Model Structure*". When LPM is in Basic Mode, the selection Button has the caption "->Advanced" to go to advanced mode and the opposite when LPM is in Advanced Mode.

5.3.1.2 Basic Mode

Few choices are available for the Plant Engineer, in order to simplify the procedure for not expert users.

ABB. All rights reserved

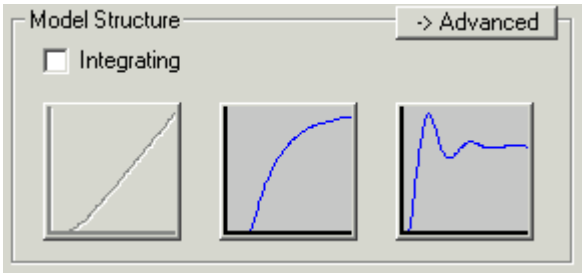


Figure 71 - Model Structure form

Taking into account Figure 71 - Model Structure form, in Basic Mode three models structure are available for identification and they can be selected by means of three buttons. The first button can be chosen only when the check-box integrating is selected, on the contrary the second and third button can be pressed only when the check-box is false.

Starting from the left of Figure 71 - Model Structure form, the model characteristics associated with the three buttons are:

- Integrating model – this is the typical behavior of processes like level in a tank, where, if the control valve is opened, the level continuously increases without reaching a steady state condition, as long as the valve is kept opened.
- First order – this is a typical behavior of many process, where, if the Control Output makes a step, the Process Variable moves from the starting point to a new steady state condition with a continuous trend line without any step or oscillation of the PV.
- Second order with delay – these processes behaves in a similar way of "First Order", but they reach the new steady state condition with some oscillation.

Each of these model structures takes also into account of a dead-time between the changing of CO and starting time when the PV changes.

Pushing one of the three buttons, the process structure is selected and the relevant parameters calculation is performed.

5.3.1.3 Advanced Mode

The identification in Advanced Mode allows using complex model structure and, if necessary, to fully control all model details. This Mode is more complex to use, but it allows reaching high model accuracy.

- If the Manually Specified check box is unselected the "best" model structure is automatically selected
- If the Manually Specified check box is selected, the user can select the model structure elements manually from the drop-down lists.

Figure 72 - Selecting the Transfer Function shows how the drop-down menus allow the selection of various terms in the transfer function model. In the numerator part, the first box allows to select whether the transfer function will have a zero or not, and in the second box you select whether it

ABB. All rights reserved

should have a dead-time or not. In the denominator part, a drop-down menu allows to select the order of the denominator, up to third order. The second order factor can be specified as well damped (two separate time constants T_1 and T_2) or resonant with relative damping ζ. (A short introduction to transfer functions is given in Appendix B – Transfer Functions).

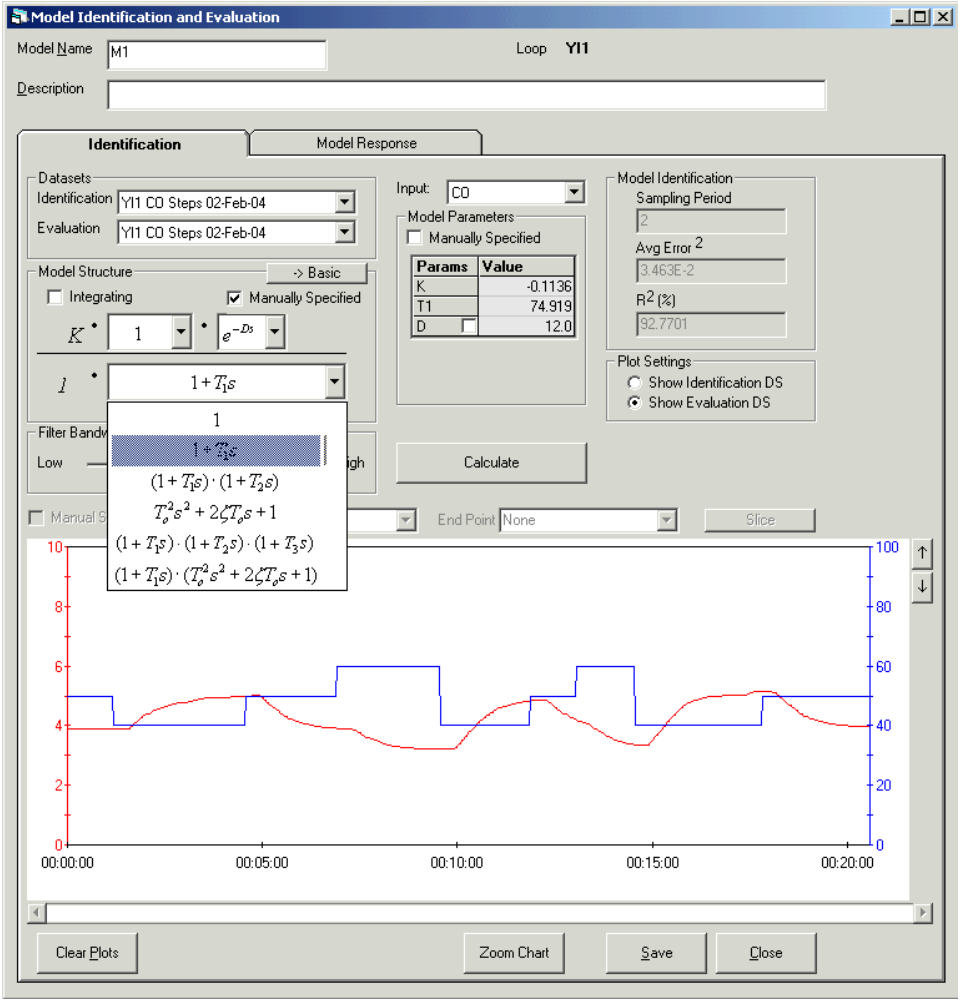


Figure 72 - Selecting the Transfer Function

On the top-left corner of the frame a checkbox allows to specify whether to have a pure integrator or not in the model. Figure 73 - Model Identification for Integrating Loops shows how the form changes when the checkbox “Integrating” is selected.

ABB. All rights reserved

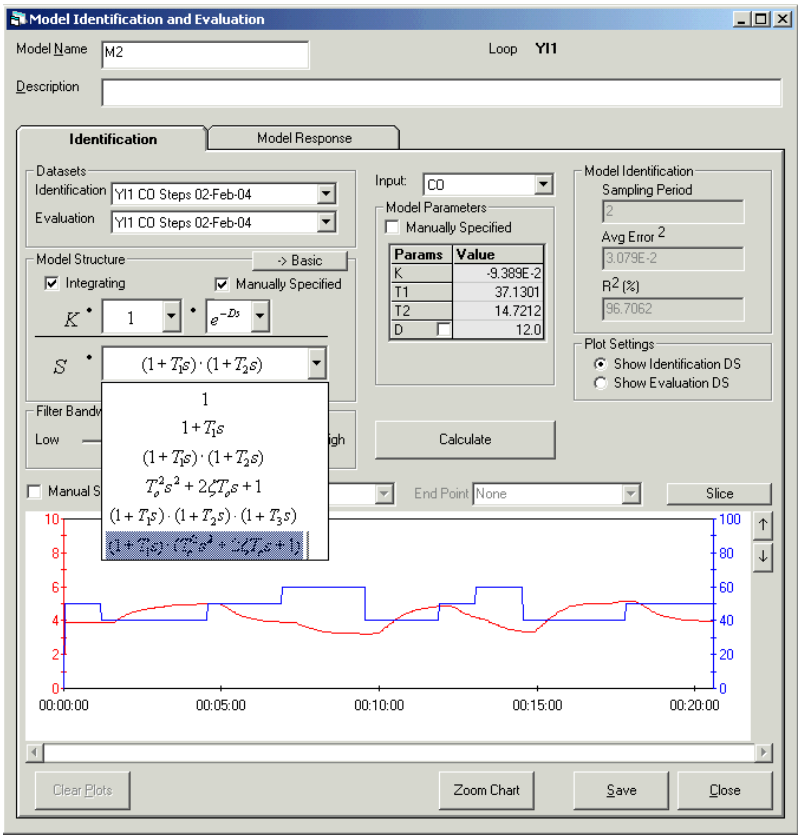


Figure 73 - Model Identification for Integrating Loops

A third frame on the right labeled "*Model Parameters*" changes dynamically in order to display the parameters calculated by LPM for the actual Model Structure identified or selected.
If the deadtime is part of the model structure, it is always possible to enter it manually after checking the box next to the deadtime (D) model parameter.
If the D checkbox in the Model Parameters frame is not enabled, then the model identification algorithm identifies the deadtime.

As a further specification the user may indicate the level of noise filtering to include when building the model. In particular, a low-pass filter is applied to smooth the raw data. The user may adjust the "*Filter Bandwidth*" slider to define the bandwidth of this filter. The High setting implies no filtering (i.e. the identified model tries to fit data in the entire frequency range), while moving toward the low setting implies a smaller bandwidth or more filtering to smooth the data. The filtering can help reduce high frequency noise to obtain more accurate models.

5.3.1.4 Defined Model assessment

Below the "*Model Parameters*" frame, the Calculate button allows the user to launch the model identification for the information provided. After clicking the calculate button a process model is built and its performance may be assessed in the following two ways:

ABB. All rights reserved

- **Numerically**: through the "*Model Identification*" frame, placed on the far right. Here both the Squared Average Error ("*Avg. Error*^{2}") and the R-squared factor ("*R^2 (%)*") are shown to provide a quantitative assessment of the model fit.

  The Squared Average Error should be as low as possible and it is defined as:

$$AvgErr^2 = \frac{1}{T}\int_0^T [\hat{y}(t) - y(t)]^2 dt = \frac{1}{N}\sum_{k=1}^{N} [\hat{y}_k - y_k]^2$$

  .

  The R-squared factor is in percentage and should high as possible. It is defined as:

$$R^2 = \left[1 - \frac{var(y - \hat{y})}{var(y)}\right] * 100$$

  An R^2 value above 90-95% is usually an indication of a good fits.

- **Graphically**: comparing the recorded PV data with the PV prediction obtained by driving the model with the input data. This graph is shown in the trend area at the bottom of the window. Please note that the two radio buttons placed in the "*Plot Settings*" frame allow the user to switch between the Identification and Evaluation datasets. The PV predictions for a model are only shown when selecting the Evaluation dataset for display.

Figure 74 - Graphical Assessment Example shows an example of the graphical assessment tool. The blue line represents the CO plot, the red one the PV plot and the green one the Model Output.

ABB. All rights reserved

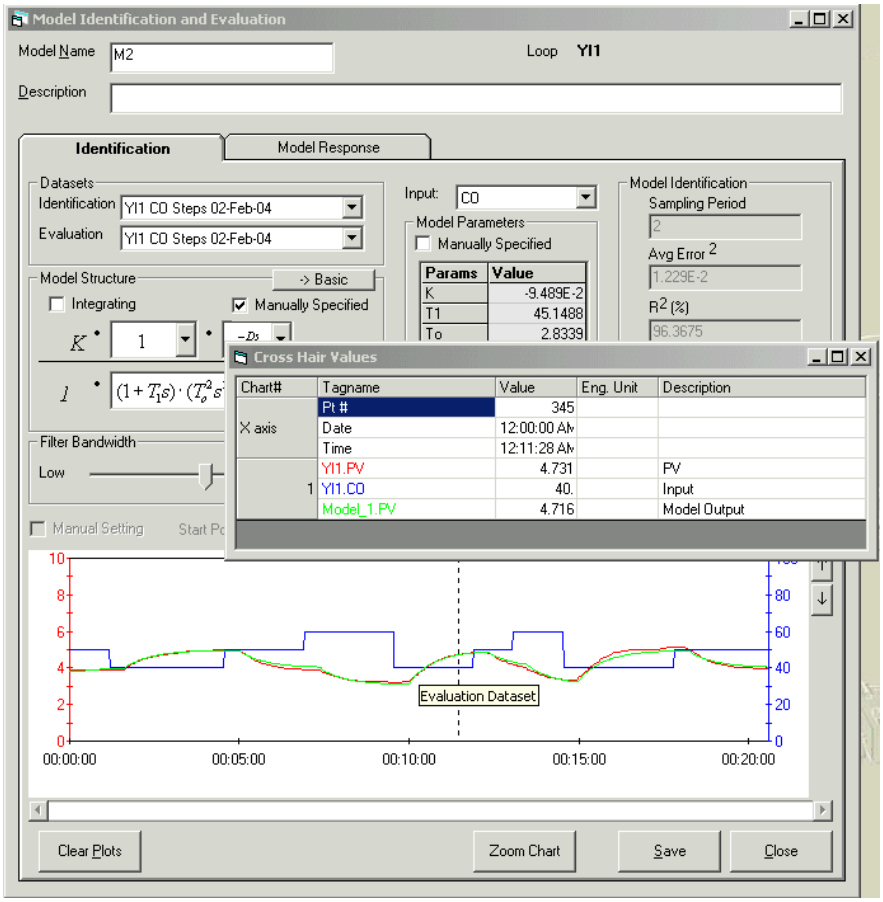


Figure 74 - Graphical Assessment Example

The colors may be checked in the Cross Hair window, which can be brought up using the context menu (right mouse click).

In case more than one model has been developed, it is possible to plot and compare more model outputs. Figure 75 - Zoomed Image of the Model Plots shows a zoomed image of the plots. The trend plot allows graphical assessment one of multiple models, simultaneously. Each time the Calculate button is pressed; the results are calculated and displayed in the plots.

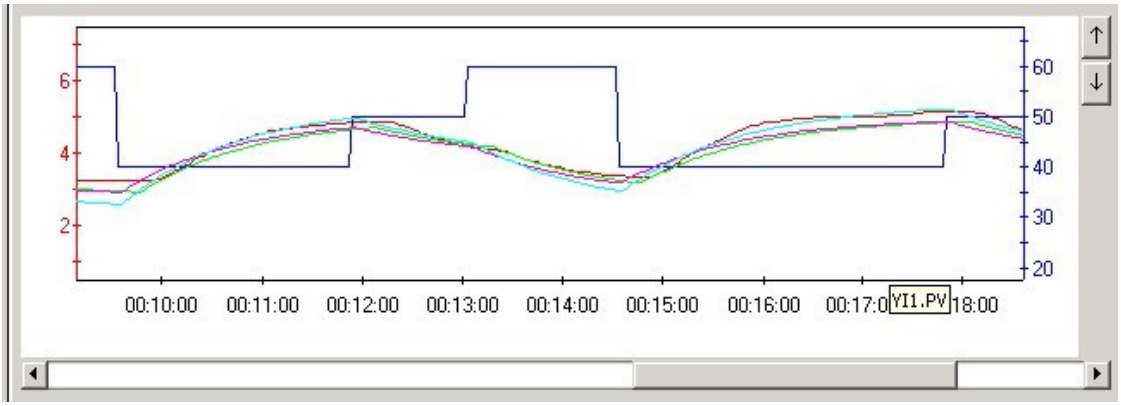


Figure 75 - Zoomed Image of the Model Plots

ABB. All rights reserved

At the very bottom of the form, there are three buttons:

- **Reset Plot**: Click to clear the diagram of all simulated process values.
- **Save**: close the model identification session, saving the results (i.e. model structure, parameter values and the numerical values quantifying the model fit)
- **Close**: close the model identification session, without saving the results

5.3.2 Model Response Tab

The Model Response tab allows the user to assess additional model characteristics. After having completed a model identification step the user may assess two different Model Evaluation Plots by clicking on the Model Response tab.

Figure 76 - Main Evaluation Screen – Time domain shows the main Evaluation screen in case "Step Response Plot" is selected. This diagram shows the PV behavior in case CO performs a unit step.

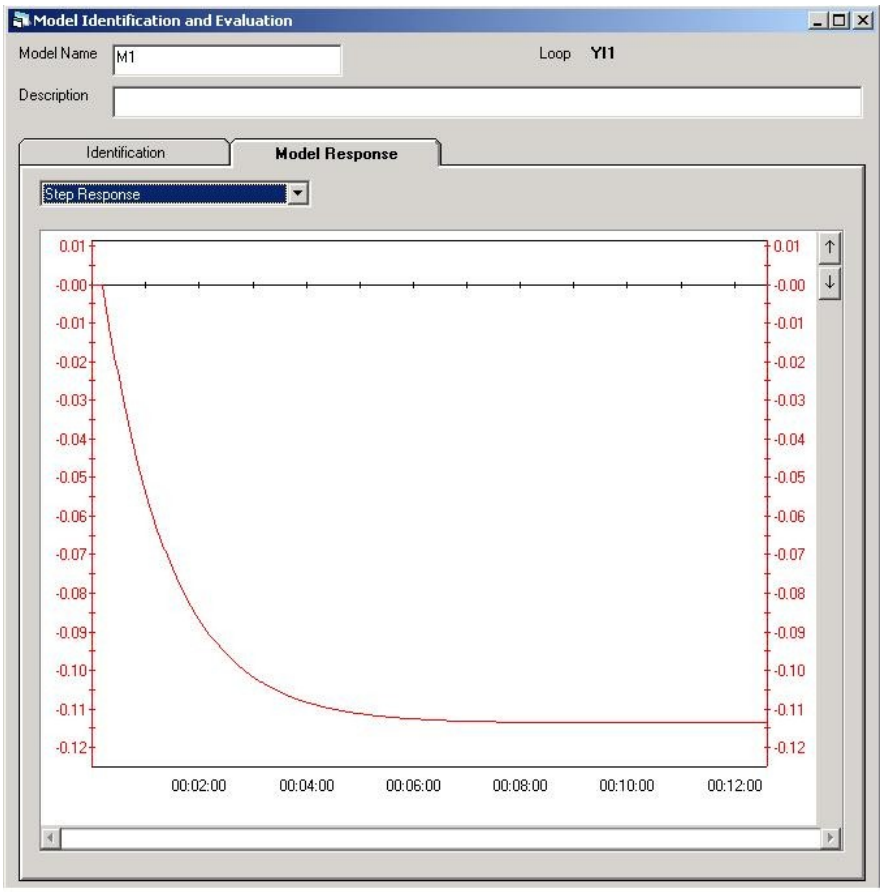


Figure 76 - Main Evaluation Screen – Time domain

Through the drop-down box to the user may select Bode Plot option. This consists of two charts:

- **Amplitude** (top diagram): Shows the amplitude gain for the transfer function from the model input (control output) to the process value, plotted against frequency.

ABB. All rights reserved

- **Phase** (bottom diagram): Shows the phase angle delay for the transfer function from the model input (controller output) to the process value, plotted against frequency.

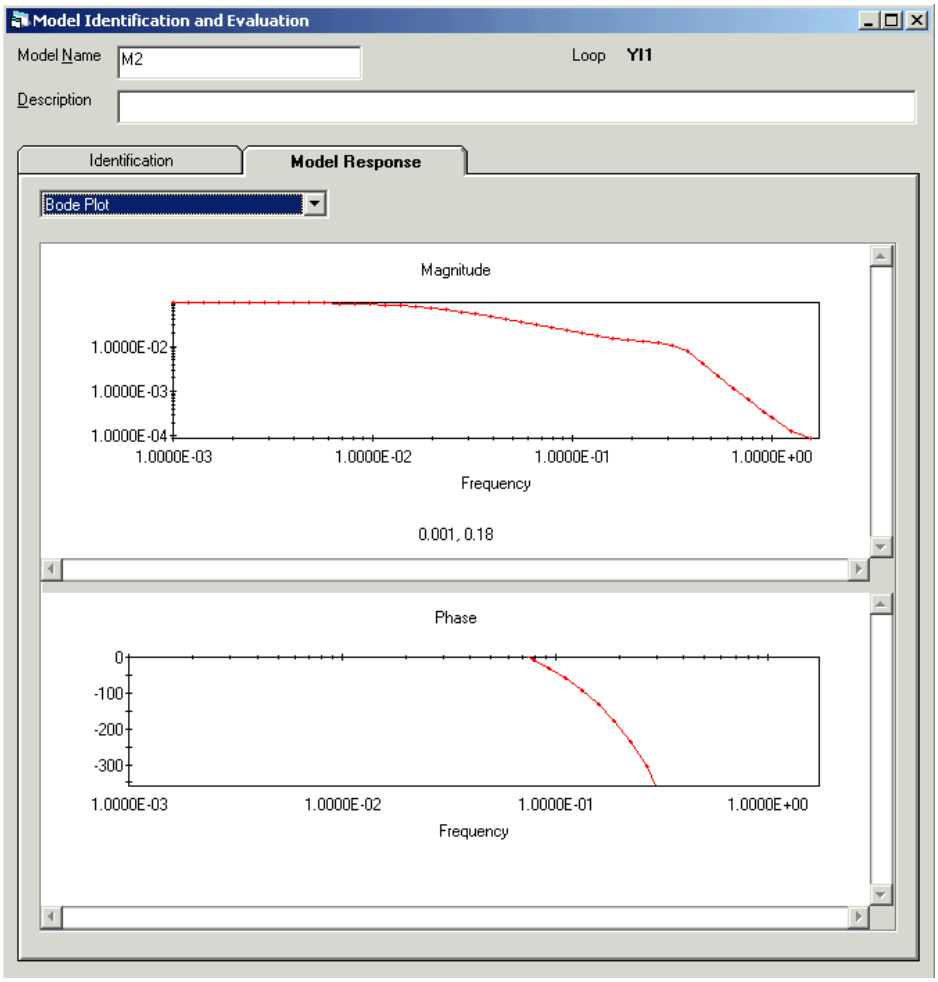


Figure 77 - Main Evaluation Screen – Frequency domain

ABB. All rights reserved

In summary, to build and evaluate a process model as part of model identification:

- Select the desired datasets (it is possible to select different datasets for identification and evaluation);
- Select the input variable for the model (CO or FF);
- Select the desired model structure, or let the model identification algorithm determine it;
- Select the desired level of noise filtering through the slider;
- Click Calculate to identify a model and display simulation and frequency response plots;
- Assess the results both on the **Identification** tab (numerically and graphically) and on the **Model Response** tab (Step Response and Bode Plot)

If the evaluation is unsatisfactory, the user can try another model structure, fix or release the dead-time, or change the filter bandwidth. If satisfactory the user may move to the Tuning section.

ABB. All rights reserved

5.4 *Controller Tuning*

Tuning seeks to find PID controller settings that give the desired performance and robustness to the control loop. Here it is assumed that a model of the process is already available, usually obtained by identification.

The user specifies the desired control behavior using one or more sliders, or by entering the corresponding numeric values, for one or more variables which define the trade-off between performance and robustness for the chosen tuning method. Performance is related to how fast a control loop counteracts set-point changes and load disturbances. Robustness indicates the insensitivity of the controller to changing conditions, such as a control valve with different behaviors at different operating points.

Manual Tuning is also available to compare tuning parameters automatically defined by LPM with Plant Engineer selection.

The resulting closed loop properties are illustrated by step and frequency response graphs. Steps are applied to the setpoint and an input disturbance. Some performance measures, quantifying the quality of the resulting control, are also presented.

The tunable parameters depend on the controller being treated. For instance, the user may specify not to use a full PID, but restrict the class to, for example, PI or PD (Manual tuning). The available parameters also depend on which control system is used. In some systems the derivative filter is freely tunable, in others not, and some controllers have a setpoint weighting factor in the proportional part, while others do not. In any case, the results are displayed according to the actual structure of the selected controller, so that it is presented to the user in the expected format, e.g. showing the applicable part of the proportional band or gain used, resets per minute or reset time, etc.

LPM offers the following tuning methods:

- Manual
- Lambda tuning (PI)
- Dominant Pole Placement, PI
- Dominant Pole Placement, PID
- IMC tuning PI
- IMC tuning PID
- Lambda-Smith

...where:

- *Manual* means specifying the control parameters manually,
- *Lambda PI* involves specifying the performance/robustness trade-off by way of the time constant (λ) of the entire closed loop,
- *Dominant Pole Placement* involves specifying the performance/robustness trade-off by way of two variables, Relative Damping of Dominant Poles (ζ) and Max. Permitted Dominant Frequency (ω_{MAX}).
- *IMC* tuning is based on the ideas of *Internal Model Control*, here adapted to allow the specification of either the maximum sensitivity, or the closed loop time constant like in lambda tuning.
- *Lambda Smith* is a special method to tune ABB Bailey controllers using "Inferential Smith Controller" algorithm instead of PID regulator (see 14.3.2.3 for details).

ABB. All rights reserved

A suitable procedure for tuning is suggested by the following sequence of subheadings, corresponding to the tabs of the Controller Tuning form:

To add a new Tuning definition, right click on Tuning in the Tree and select **Add Item** (see Figure 78 - Add Tuning) or press the relevant menu button.

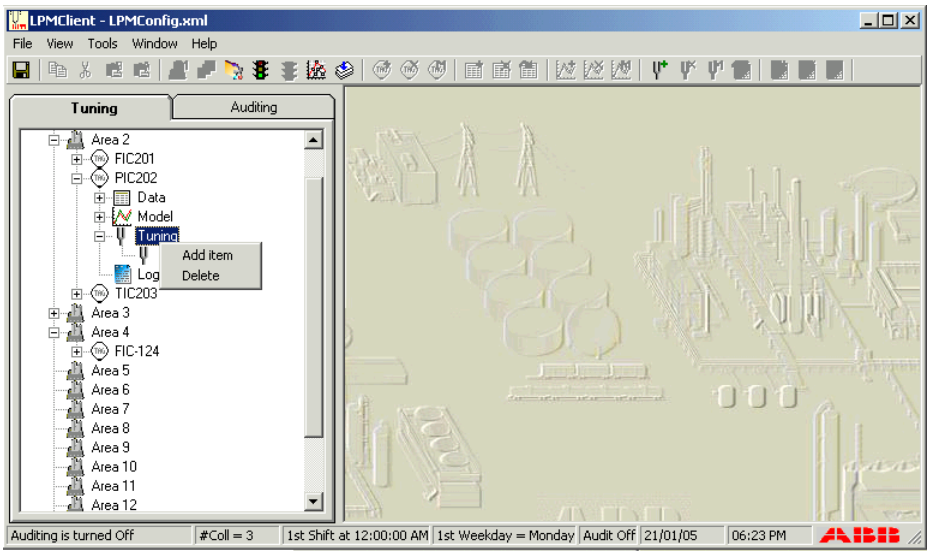


Figure 78 - Add Tuning

This will open a dialog box similar to the one shown in Figure 79 - Add Tuning Dialog.

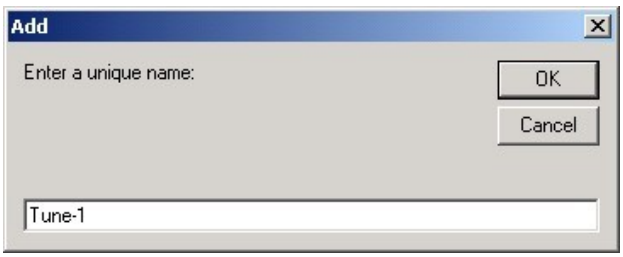


Figure 79 - Add Tuning Dialog

This action will create a new item under the Tuning branch. Right clicking on the new item it is possible to operate on the selected item. Select Tune, or double click the tuning item, to open a tuning definition (Figure 80 - Tuning Operation).

ABB. All rights reserved

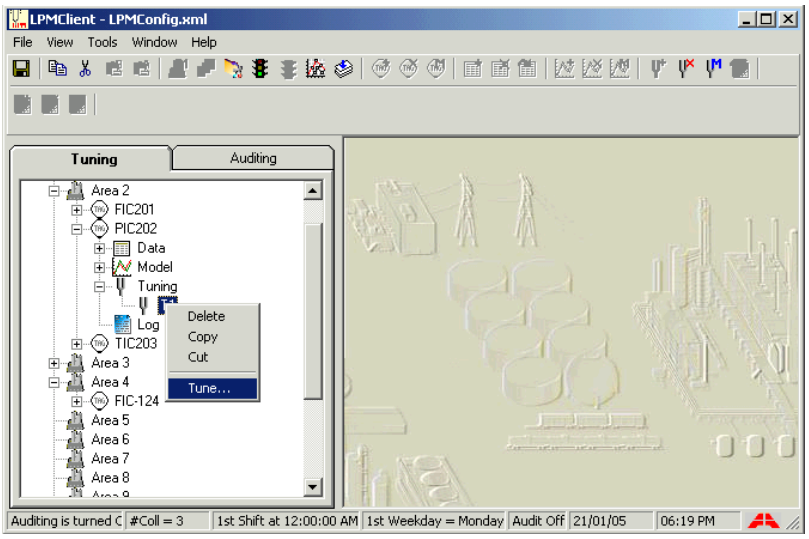


Figure 80 - Tuning Operation

The user will access the main Tuning form (see Figure 81 - Main Tuning Screen).

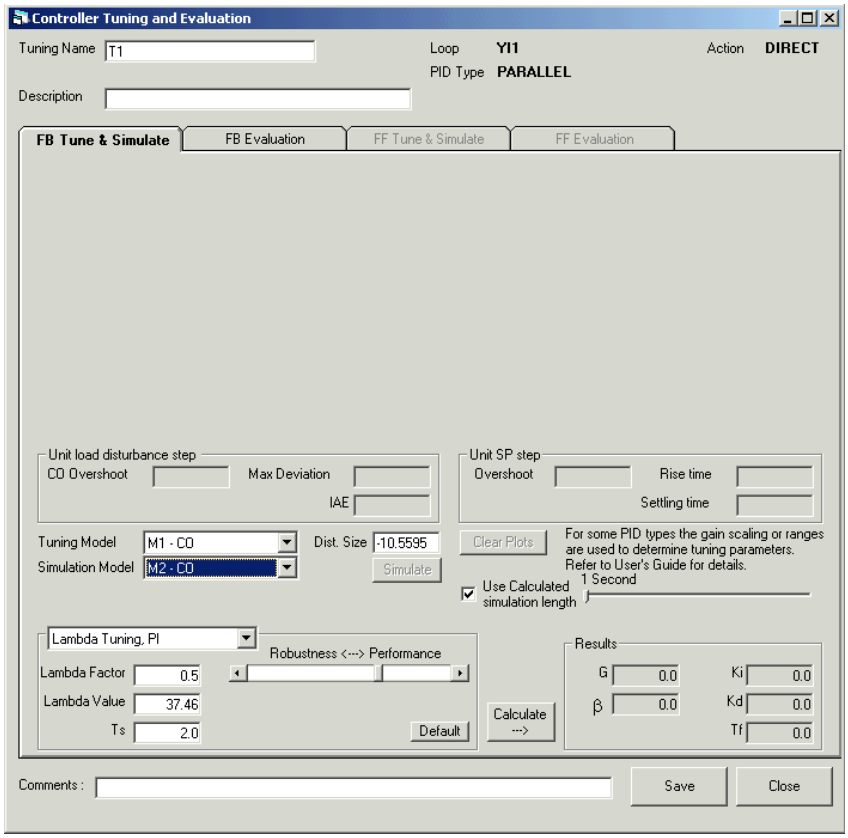


Figure 81 - Main Tuning Screen

The top part of the form is devoted to some descriptive information: Tuning Name and Description can be accessed and edited by the user while Loop Name and PID Type are automatically set according to the general selection defined in the Loop Configuration form.

ABB. All rights reserved

As shown in Figure 82 - Main Tuning Screen for Outer Cascade Loop, an additional drop-down menu appears in the Top part of the form if the analyzed loop is an Outer loop of a cascade configuration. This drop-down menu is used to select the Tuning set of the internal loop. See chapter 5.6 for details on tuning of Cascade loop.

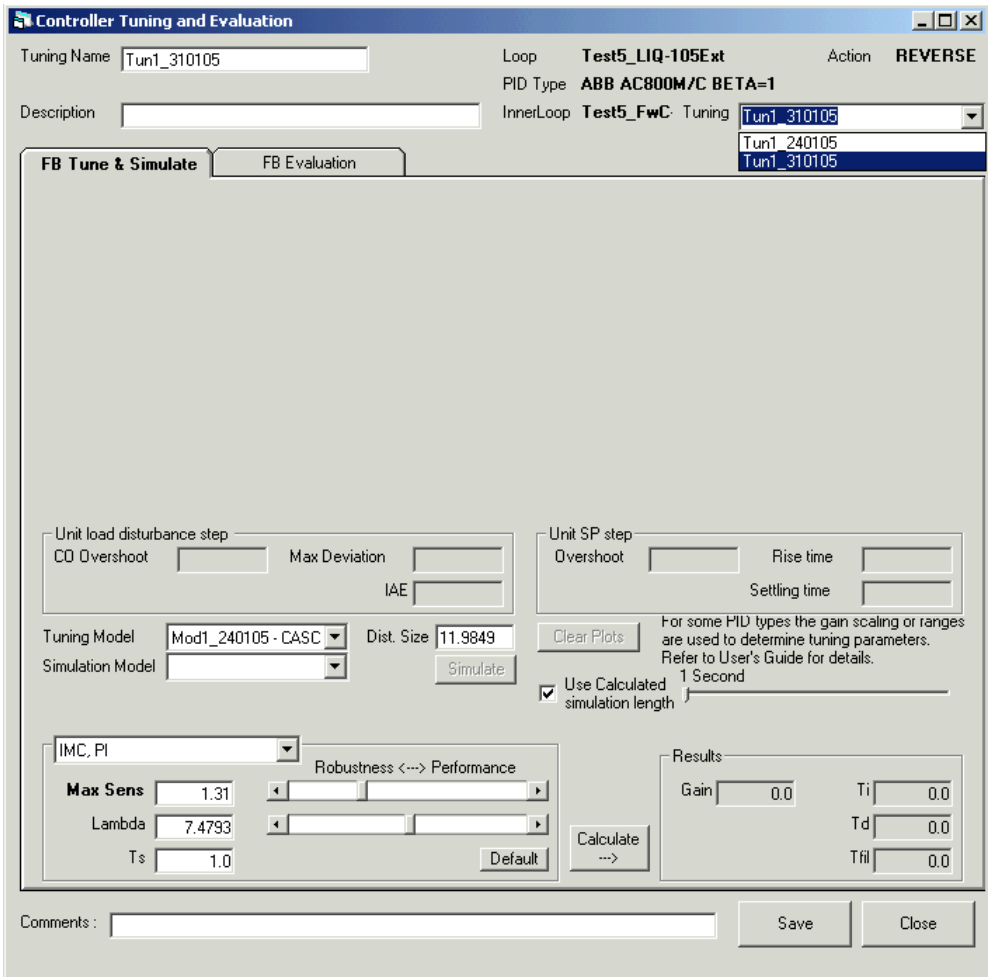


Figure 82 - Main Tuning Screen for Outer Cascade Loop

The lower part of the form is made up of four different tabs:

- The FB Tune and Simulate Tab;
- The FB Evaluation Tab;
- The FF Tune and Simulate Tab;
- The FF Evaluation Tab.

5.4.1 FB Tune and Simulate Tab

The FB Tune and Simulate Tab are used to define the feedback tuning specifications and launch the actual tuning calculations. The upper part of the form is devoted to the display of the main results, while the lower part is where the user must select the method and parameters.

ABB. All rights reserved

Tuning uses models to capture process behavior and on the basis of model characteristics best PID parameter are defined. It is possible to select two different models for Tuning purpose: “Tuning Model” used to calculate best Tuning Parameters and “Simulation Model” to validate the calculated PID parameters versus a different model. The selection among the different model identified is carried out by means of drop-down menus listing all available models for the given loop.
In this section (FB Tune and Simulate) only the models with the “–CO” suffix will be shown since these models correspond to the control output signal.

Below the model selection, a second drop-down menu allows selecting the desired tuning method (Figure 83 - Tuning Method Selection).

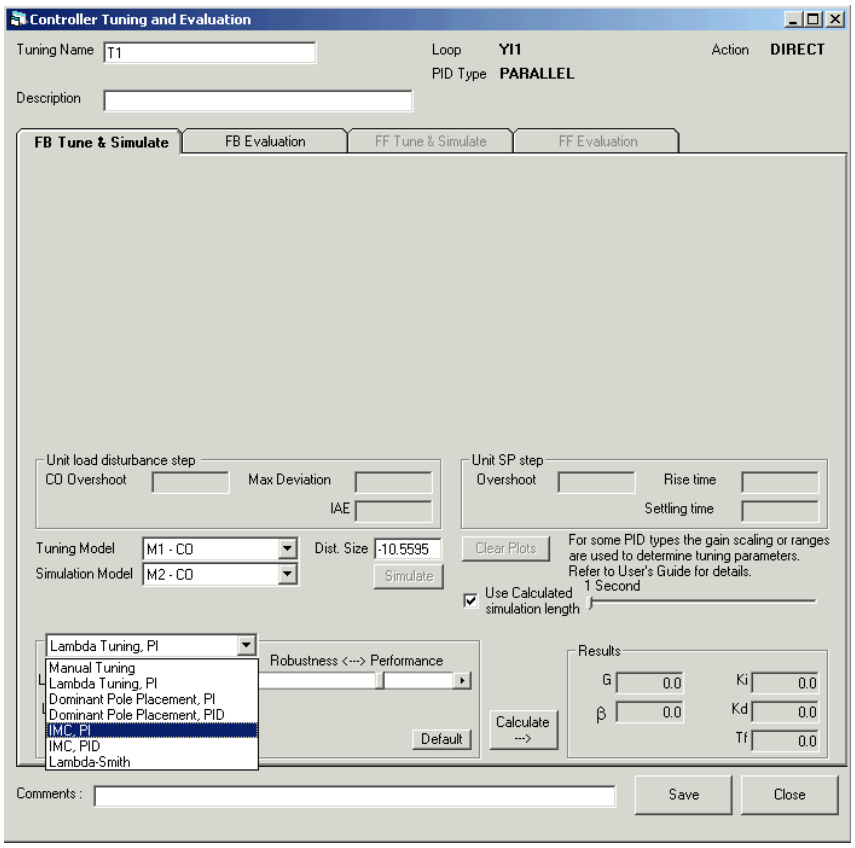


Figure 83 - Tuning Method Selection

LPM includes the following tuning methods:

- Manual Tuning
- Lambda Tuning PI
- Dominant Pole Placement, PI
- Dominant Pole Placement, PID
- IMC PI
- IMC PID
- Lambda-Smith
- Integral Time Absolute Error (ITAE)

ABB. All rights reserved

After having selected the tuning method, the user may start exploring the possible tuning alternatives, by simply clicking the Calculate button.
This will launch the tuning calculations and provide the optimal tuning parameters, which will be shown in the frame labeled "**Results**" near the bottom-right corner. Note that LPM displays the tuning results in the proper format of the specified controller (indicated in the Loop Configuration form) to simplify the loading of the optimal tuning parameters.
In Figure 84 - Optimal Tuning Parameters Calculation for example, the results are displayed in terms of the tuning parameters for the ABB Bailey FC156 PID controller (spec. S12, S13, S14, S15).

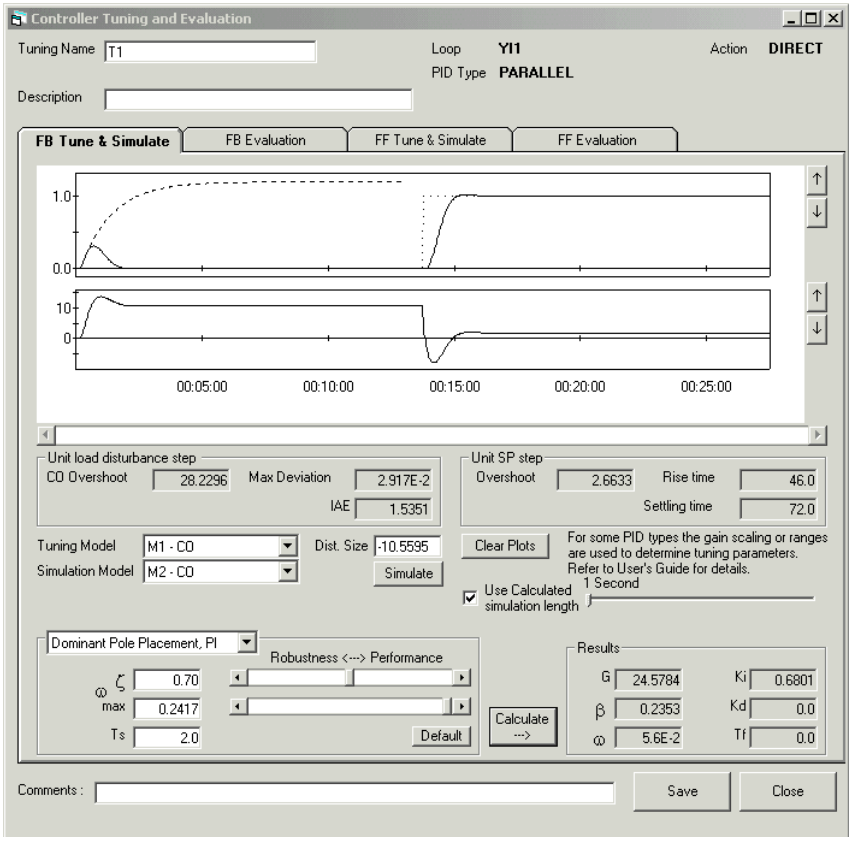


Figure 84 - Optimal Tuning Parameters Calculation

Together with the tuning parameters displayed in the "Results" frame, clicking the Calculate button will also update the charts in the upper part of the form.
These trends are simulations of the control loop, tuned with the newly calculated parameters, against two typical test cases:

- The left part of the chart displays the PV (upper chart) and the CO (lower chart) response to a unit **load disturbance** step.
- The right part of the chart displays the PV (upper chart) and the CO (lower chart) response to a unit **setpoint** step.

The size of the load disturbance step applied in the simulation is shown beside the Model selection (**Dist. Size**). The disturbance amplitude is automatically selected by LPM, in order to have the amplitude of the PV response roughly similar to 1 for better visualization purpose. However it is also

ABB. All rights reserved

possible to alter the size from the provided default, and repeat the simulation, if you like. It will not influence the numerical values discussed below, since they are always rescaled to unit step size.

After tuning by pushing "**Simulate**" button, it is possible to validate calculated PID setting versus a model different from the one used for the tuning itself. This allows to evaluate controller robustness to model errors or different working conditions.

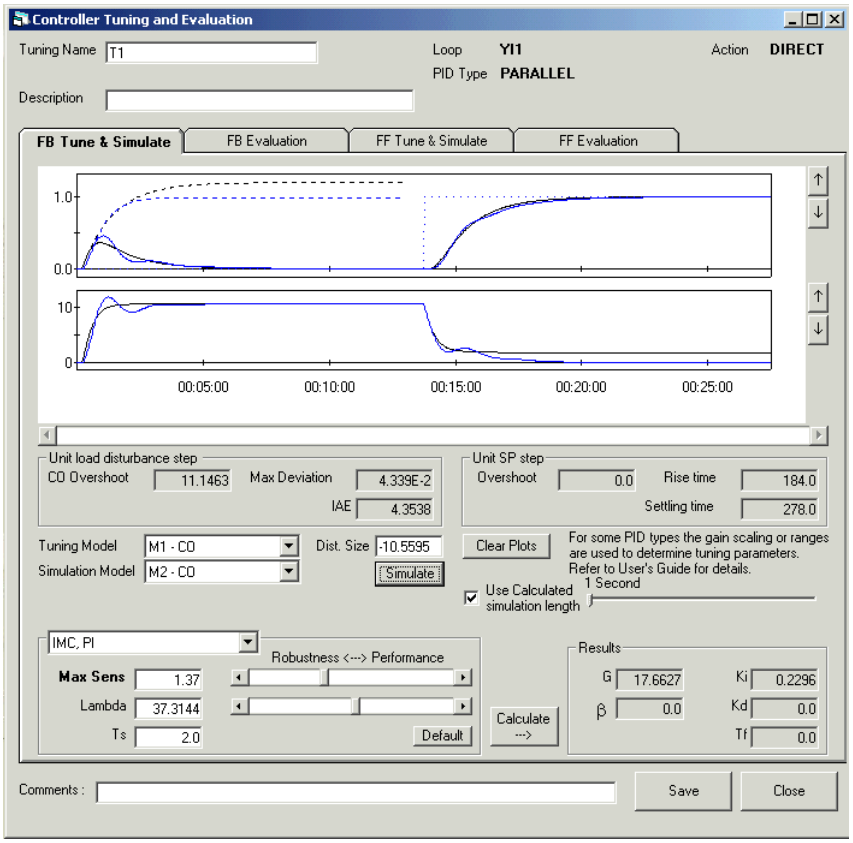


Figure 85 – Tuning with "Tuning Model" and "Simulation Model"

Simulation time is automatically calculated, but it can be also manually defined by moving the slider activated by the check box "Use Calculated Simulation length".

Below the charts there are two frames containing numerical performance and robustness statistics specific to the test cases.

- In the "**Unit Load Disturbance Step**" frame, three parameters are reported:
  1. CO overshoot = the maximum overshoot of the Controller Output in percentage
  2. Max. deviation = maximum absolute value of PV error
  3. IAE = Integrated Absolute Error, defined as:

ABB. All rights reserved

$$IAE = \frac{\int |e|\, dt}{D} = \frac{\sum_{k=1}^{n} |e_k| * T_s}{D}$$

where: D is the disturbance size
Ts is the sampling period of the controller

- In the "**Unit SP Step**" frame, three parameters are reported:
  1. Overshoot = the maximum overshoot of the Process Variable in percentage
  2. Rise time = the time in second needed to the Process Variable to go from 10% to 90% of the whole response to the SP step
  3. Settling time = the time in second needed to the Process Variable to enter and stay in the region within ±5% of the final PV value

In the bottom-left corner it is possible to select which tuning method to use. Tuning is always a trade-off between robustness and performance and the present choice of robustness/performance level is illustrated in the area labeled "**Robustness <--> Performance**" (unless you have chosen Manual Tuning). Depending on which method has been chosen, one or two sliders are available: moving the slider to the right the controller tuning will be more "aggressive" to increase performance, while more sensitive to unmeasured disturbances and/or model mismatch. Moving to the left a premium will be placed on model robustness. It will be able to cope with larger deviations between model and reality, but it will be a bit slower in counteracting disturbances. The actual value of the parameters corresponding to the slider position (for example in case of Dominant Pole Placement ζ, ω_{MAX} T_s of Figure 86 - Performance and Robustness Parameters) are also explicitly reported in this frame, together with a button that allows the user to restore the default values at any time.
Entering numerical values will move the slider as well, and vice versa. It is easy to verify the effect of the choice in the simulated plots.
The remaining input to the tuning calculations, available in this area, is the PID controller cycle time Ts. This makes it possible to evaluate whether there is reason to change it from what it has already been configured to be in your control system.

ABB. All rights reserved

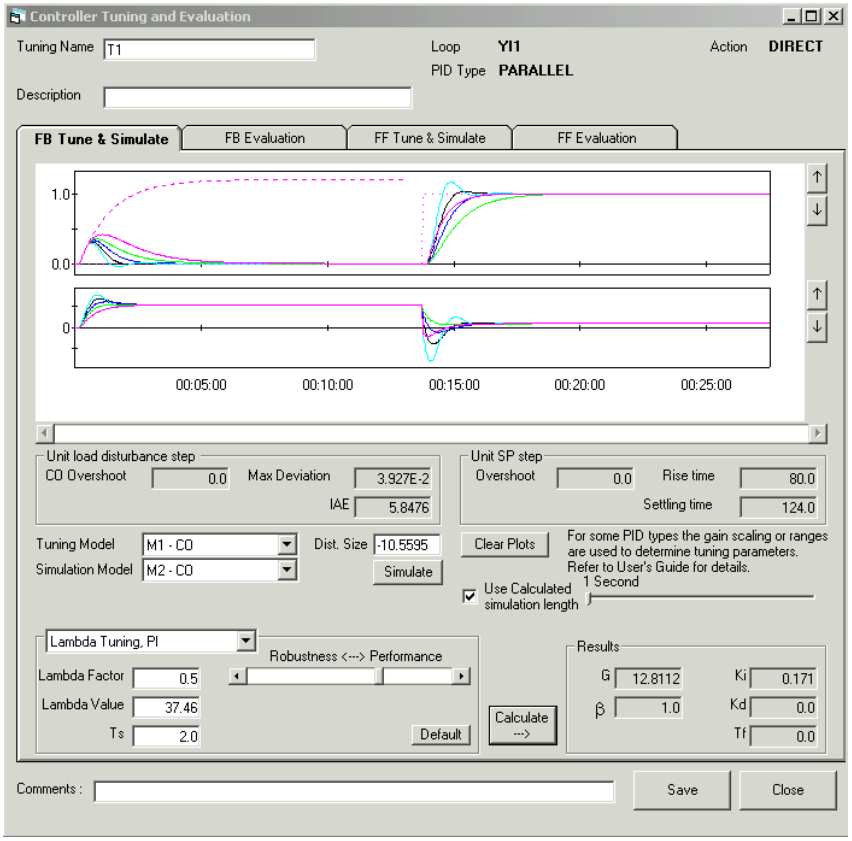


Figure 86 - Performance and Robustness Parameters

On the right side of the form, the button “Clear plots” is available to remove all traces from the charts displayed in the trend area.

In the following some basic information about each tuning method specific parameters are provided:

5.4.1.1 Manual Tuning

The manual tuning method is used to compare LPM choice of tuning parameters with plant engineer selection. This method can start from the setting obtained by any of the automatic tuning methods, so you only need to change the parameters you want to. When manual tuning is selected, a Copy button appears that allows you to copy the previously obtained tuning result to the parameter entry fields, where you can adjust them manually.

- Gain: Enter the controller gain.
- T_i: (only available for PID-controllers of the ideal or series type) Select the small check box to the right of the Ti box to enable it and then enter the integration time. Clear the check box to disable the integration part, which means that you get P or PD control.
- Ki: (only available for PID-controllers of the parallel type) Enter the integration parameter. The value '0' corresponds to P or PD control.
- β: Enter the set-point weighting factor to use in the proportional part (see Appendix with PID types)
- T_d: (only available for PID-controllers of the ideal or series type) Enter the derivative time constant. The value '0' corresponds to P or PI control.

ABB. All rights reserved

- K_d: (only available for PID-controllers of the parallel type) Enter the derivative time constant. The value '0' corresponds to P or PI control.
- T_f: Enter the derivative filter time constant.

Note that the actual tuning parameters available for editing depend on the specified PID controller type (see Appendix E).

ABB. All rights reserved

5.4.1.2 Lambda Tuning

For Lambda tuning, the user specifies either the desired time constant of the closed loop or the speed ratio (ratio of closed loop to open loop time constants). When changing one parameter, the other parameter is automatically recalculated. The slider indicates how to change the values to increase robustness or performance. This method is restricted to first order plus dead-time models (first order being either pure integration or a time constant).

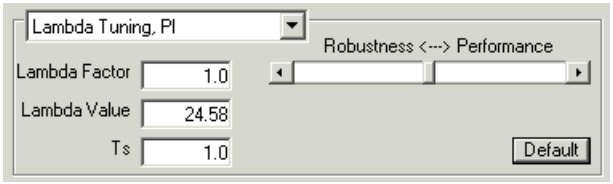


Figure 87 – Frame for Lambda Tuning

- Lambda Factor: this is a speed ratio or slowdown factor. If greater than one, the closed loop is made slower than the open loop. If less than one, it is made faster.
- Lambda Value: this is the desired time constant of the closed loop in second.

Robustness <---> Performance:

- One slider: Move the slider to select the desired robustness/performance trade-off.

When changing one parameter, the other parameter is automatically recalculated, and the slider is moved. When moving the slider, both parameters change.

- Default: Click to restore default settings.

5.4.1.3 Dominant Pole Placement PI or PID

Dominant pole method considers the resulting close loop system as a second order system. The user defines the damping ratio (ζ) and maximum frequency (ω_{max}) of the second order close loop model. These parameters can be set using sliders or by manually entering values. The sliders indicate how to change the values to increase robustness or performance.

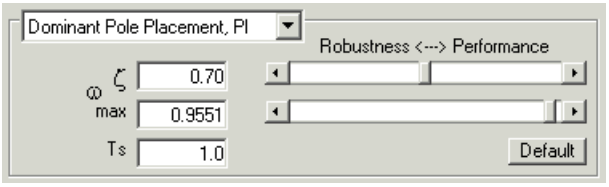


Figure 88 – Frame for Dominant Pole Placement Tuning

- ζ: this is the desired relative damping of the closed loop.

ABB. All rights reserved

- ω_{max}: this is the maximum allowed dominant frequency (rad/s).

Robustness <---> Performance:

- Top-most slider: Move the slider to select the relative damping (ζ).
- Bottom-most slider: Move to select the maximum allowed dominant frequency (ω_{max}).

The two parameters are independent.
Relative Damping (ζ) directly acts on the trade-off between performance and robustness. Lower damping ζ results in short response time for close loop system, but smaller stability and robustness. Instead to obtain a close loop system more robust, it is necessary to increase the relative damping (ζ) and therefore move the top-most slider towards the right.
You would normally not be able to choose a suitable ω_{max} value directly yourself, but after you have seen what actual dominant frequency (ω) you get, you are able to make the closed loop slower and more robust by selecting a suitable lower value than that for your ω_{max}.

- Default: Click to restore default settings.

5.4.1.4 IMC tuning PI and PID

With Internal Model Control tuning (IMC) you can specify lambda just as for lambda tuning, but IMC copes with higher order models as well. In addition to that, you can instead specify the desired maximum sensitivity (Ms), and get the corresponding lambda value calculated automatically.

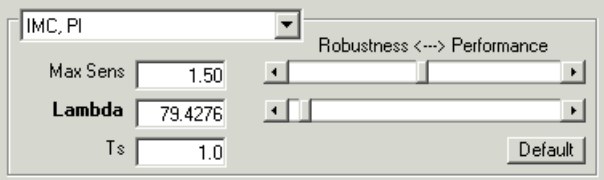


Figure 89 – Frame for Internal Model Control Tuning

- Max Sens: this defines the desired peak value of the sensitivity function magnitude;
- Lambda: this defines the lambda value, i.e. the desired closed loop time constant

Robustness <---> Performance:

- Top-most slider: Move the slider to select the maximum value of the sensitivity function (M_s)
- Bottom-most slider: Move to select the desired lambda value

M_s is always greater than 1 and in order to design stable system it is below 2. Lower values are more robust, higher less. If you specify the maximum sensitivity (either by entering a value or moving its slider), the text **Max Sens** becomes bold to indicate that this is the valid specification.
When you click Calculate, the lambda value and its slider position will be updated. Correspondingly, if you enter a lambda value or move its slider, the text **Lambda** becomes bold to indicate that the

ABB. All rights reserved

lambda value is the valid specification, and when you click Calculate, the maximum sensitivity value and its slider position will be updated.

- Default: Click to restore default settings.

5.4.1.5 Lambda-Smith

This method is designed to tune ABB Bailey "Internal Smith Controller" (ISC - Function Code 160). This controller type is suitable for process characterized by significant dead-time, because the control action is calculated using a model of the process able to predict the PV, before it is effectively measured. For details see 14.3.2.3.
Being designed for ISC controller, this method is only applicable, if this type of ABB Bailey controller is selected in the Loop Configuration. Moreover Lambda-Smith method is restricted to stable models (integrating process is not allowed)

This tuning method is very similar to Lambda method and it has the same sliders to select the compromise between Performance and Robustness.

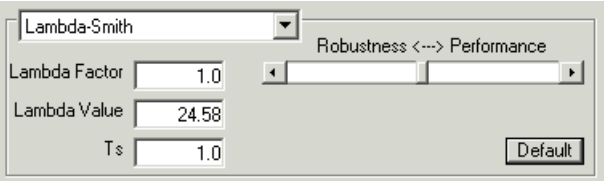


Figure 90 – Frame for Lambda-Smith Tuning

- Lambda Factor: this is a speed ratio or slowdown factor. If greater than one, the closed loop is made slower than the open loop. If less than one, it is made faster.
- Lambda Value: this is the desired time constant of the closed loop in second.

Robustness <---> Performance:

- One slider: Move the slider to select the desired robustness/performance trade-off.

When changing one parameter, the other parameter is automatically recalculated, and the slider is moved. When moving the slider, both parameters change.

- Default: Click to restore default settings.

5.4.1.6 Integral Time Absolute Error (ITAE)

ITAE can be selected from the popup list in the 'Controller Tuning and Evaluation' window. It is able to design control parameters for all model types supported by LPM.
There are two main preconditions for using ITAE design:

- the process model must be stable
- initial controller values (e.g. Kp and Tn for standard PI controller) must be set. The initial controller parameters must result in a stable close loop.

ABB. All rights reserved

There are many ways to get initial values. The simplest one is to use another tuning method to get them. Another way is trial and error or guess useable initial values that stabilize the loop.
If you have used another tuning method before ITAE, then you will have the calculated controller parameters available as initial values for ITAE.
Figure 91: ITAE tuning window shows the LPM display after changing from controller design by '*Dominant Pole Placement, PID'* using default settings, to ITAE. You see that the parameters in the 'Result' frame are editable in opposite to the other tuning methods but similar to 'Manual Tuning'. In that way you are able to set/modify initial controller values.

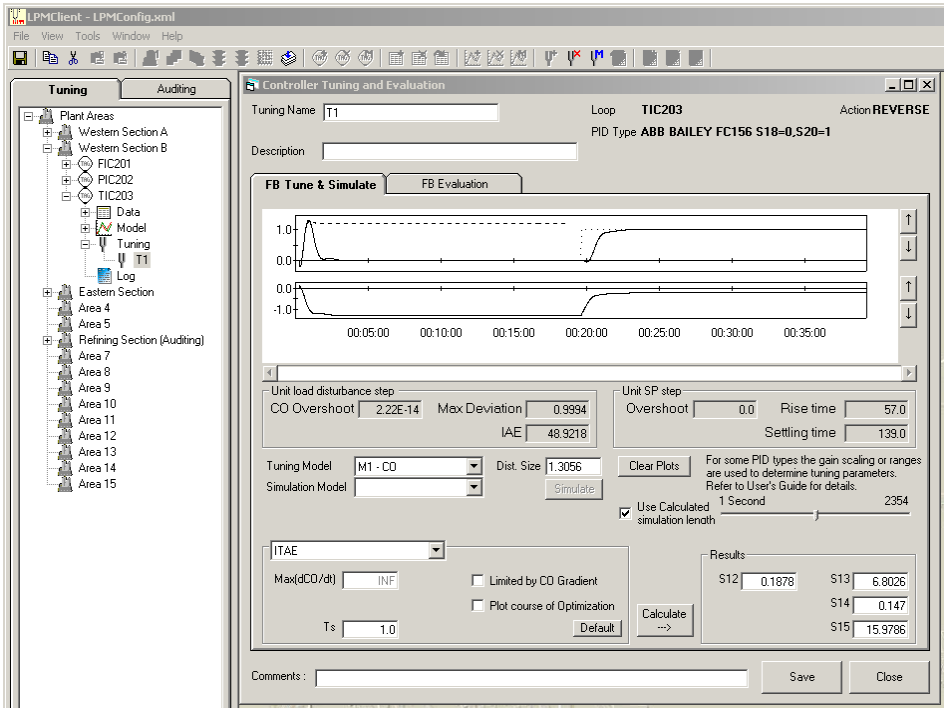


Figure 91: ITAE tuning window

The next design step is to start the calculation by clicking on the appropriate button, Figure 92.

ABB. All rights reserved

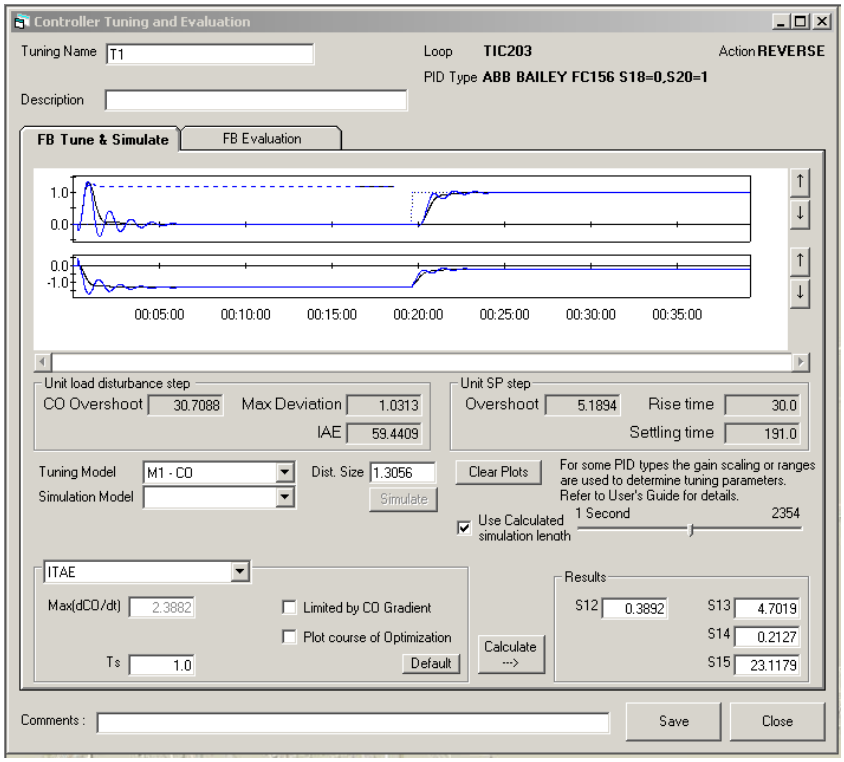


Figure 92: Control parameters calculated by ITAE

Figure 92: Control parameters calculated by ITAE shows that the resulting PID parameters and the dynamic response are different from the initial set shown by Figure 91: ITAE tuning window. The ITAE design shows overshoots in opposite to the previous design. A good damped overshoot is typically for ITAE.

The content of the field '**Max(dCO/dt)**' has also changed from default 'INF' to '2.3882'. The user will be informed about the current maximal CO gradient of the CO figure (for set point step), in that way, which is less than 2.4 $^{\%}/_{Min}$.

The user can modify this value to get a slower step response. To do that, it is necessary to select the check box '**Limited by CO Gradient'**. Afterwards, the '**Max(dCO/dt)**' value field is changed in the edit mode. Figure 93: ITAE design with limited CO gradient shows the design result for a Max(dCO/dt) = 1.5. The corresponding CO figure has only one overshoot which is higher than before.

ABB. All rights reserved

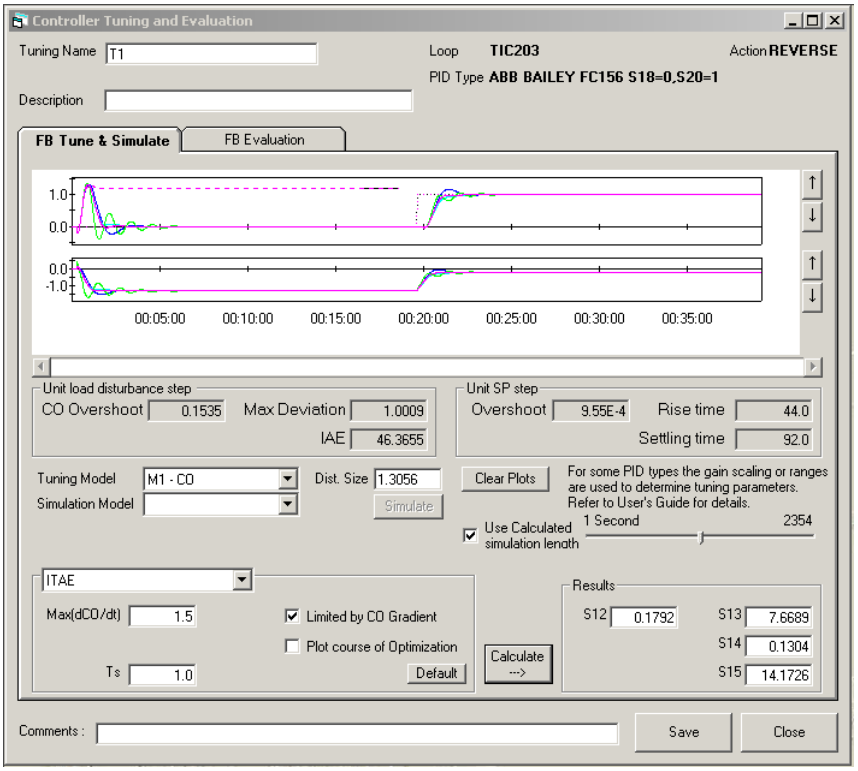


Figure 93: ITAE design with limited CO gradient

You will get a warning about wrong initial parameters if you click the calculation button without modifying the initial values from Figure 92: Control parameters calculated by ITAE. The reason for this is that the initial parameters are not able to fulfill the Max(dCO/dt)-limitation. To force this, you normally only have to reduce the proportional value in an adequate way. A tenth of S12 (0.03892) was chosen in the example here. That is a common way in control engineering to damp dynamics of simple loops. You can eliminate the CO gradient limitation by click on button 'Default'.

The check box '**Plot course of Optimization**' supports supervision of the optimization process. If '**Plot course of Optimization**' is selected the window shown by Figure 94: ITAE optimization course will be opened. It presents the optimization calculation. The window is divided into three plots, which are refreshed after every optimization step. The first plot shows the step response of the process variable. The second plot is the corresponding control value. The third plot shows the progress of the ITAE index. Every calculated index value is represented by a point. The color of the point informs about important optimization states. Green points show successful of ITAE calculations. Blue points inform about unstable loops resulting from bad controller parameters. The optimization algorithm will correct the parameters to get a stable loop in the next calculation step(s). Red points inform about Max(dCO/dt) violations. If you use the default Max(dCO/dt) = INF you will never get red points.

ABB. All rights reserved

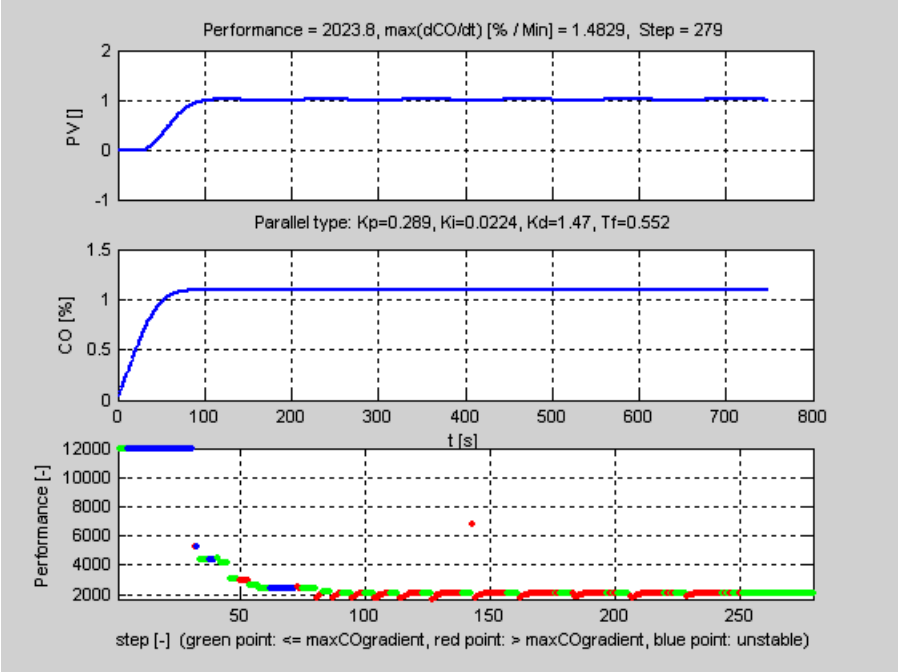


Figure 94: ITAE optimization course

A good optimization course has a lot of green points. More red and blue points than green ones are an indication for a difficult optimization process. This happens normally if you reach the loop limits. The optimization horizon will be defined by the initial values and the controller sample time Ts. Both may be helpful to modify the optimizations process.
There is further information on the optimization course window that is updated after every optimization step. Information about the current ITAE value, the current maximal CO gradient and the current optimization step number are at the top of the first plot (PV). The step number is helpful to see when the optimization did not converge. The optimization is limited to 100 steps per parameter. E.g., a PI controller has two optimization parameters; therefore, the optimization will be stopped after 200 steps if no convergence was possible. Such a result should be used carefully. At the top of the second plot, you get information about the current control parameters. These parameters are not necessarily the same as selected in the loop configuration setup. They are the parameters of the parallel controller type. All optimization calculations are based on that controller type and transformed accordingly.

5.4.2 FB Evaluation Tab

The FB Evaluation Tab (Figure 95 - FB Evaluation Tab) is devoted to investigate the tuning result in the frequency domain. The upper part of the tab presents the Sensitivity Function, which for every frequency shows the amplitude of the PV error due to a sinusoidal disturbance of amplitude. Both frequency and sensitivity function amplitude are plotted with logarithmical scale.
The maximum value (M_s) is a measure of the robustness margin, since it is the inverse of the nearest distance to the point -1 in the Nyquist curve. Beside this robustness illustration, the plot gives insight in the disturbance rejection provided by the control. It shows how much disturbances at different frequencies are damped. The lower the sensitivity magnitude value at a certain frequency, the better

ABB. All rights reserved

disturbance rejection the controller provides for disturbances at that frequency. At frequencies where the sensitivity magnitude is greater than 1, disturbances are actually amplified.

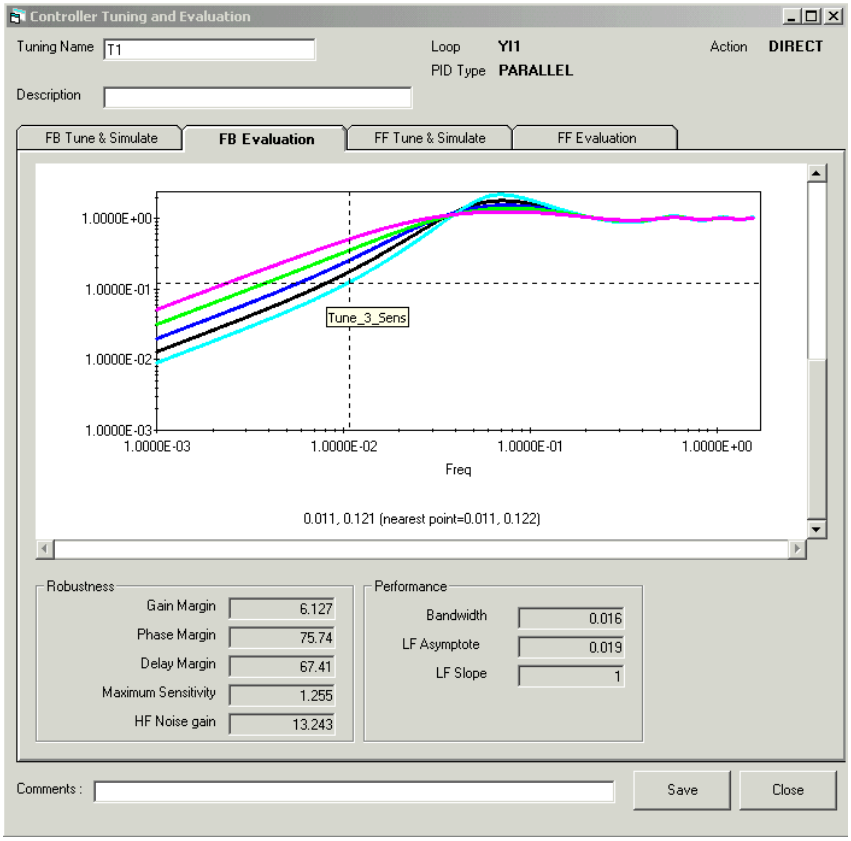


Figure 95 - FB Evaluation Tab

The lower part of the tab reports some numerical results that are grouped into two frames (see Appendix D):

- The "**Frequency Response**" frame shows the following factors:
  - Gain Margin
  - Phase Margin
  - Delay Margin
  - Maximum Sensitivity
  - HF Noise Gain (i.e. high frequency noise gain)
- The "**Disturbance Rejection**" frame shows the following factors:
  - Bandwidth
  - LF Asymptote
  - LF Slope

More than one plot can be displayed graphically as in Figure 95 - FB Evaluation Tab. More details on the evaluation parameters are available in 11.1.2.

ABB. All rights reserved

5.4.3 FF Tune and Simulate Tab

When a loop is configured for FF variables, the user can build multiple models, each one representing a single input, single output (SISO) model with either a FF variable or the CO defined as the model input. The user is responsible for collecting suitable data to use for modeling. Since the entire data collection is used to develop each model, the user should create a separate data collection with the control loop in MANUAL or AUTO with sufficient PV-independent excitation in CO, to build a model from the CO to the PV. Then, create another data collection with the control loop (without FF control active) in MANUAL mode to build a model from the FF disturbance variable to the PV during a period with sufficient excitation of the FF. An additional data collection is needed for each FF variable in order to build all the SISO models.

In the FF Tune & Simulate form, the user can select a FF variable from a drop-down menu listing all the FF variables, which are currently configured. The user must then select a model for the FF variable. Up to five FF variables can be configured and tuned for each loop, through the tabs labelled as **AUX1, AUX2, AUX3, AUX4, and AUX5** (Figure 96 - FF Tune & Simulate Tab). Each feedforward is tuned separately.

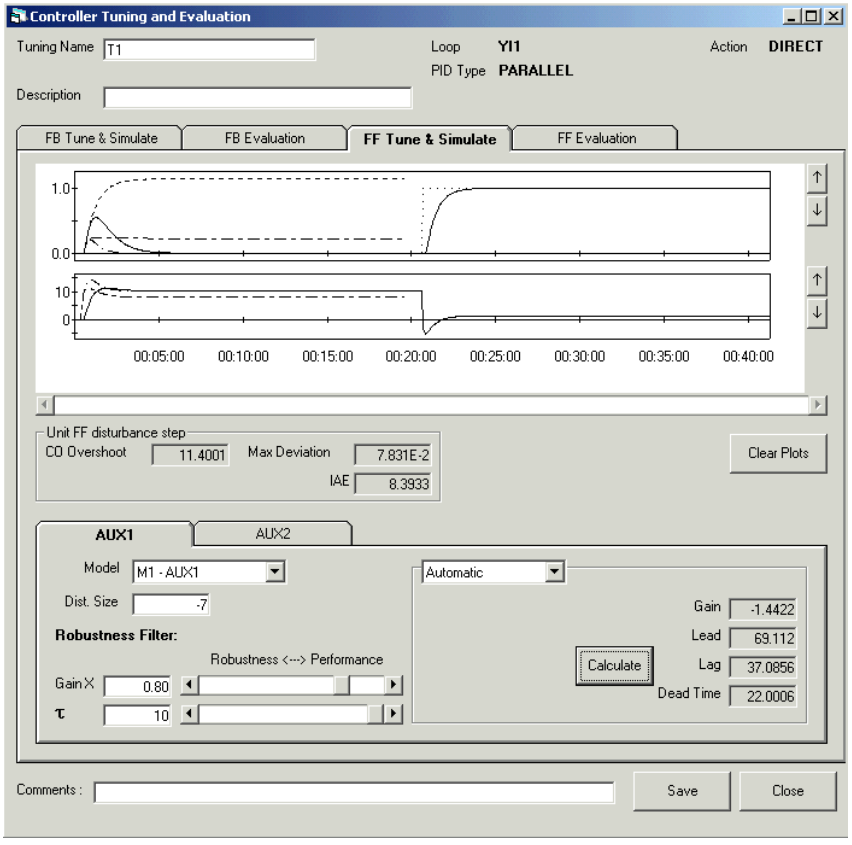


Figure 96 - FF Tune & Simulate Tab

For FF tuning, the user has two choices from the drop-down menu: Automatic or Manual.

LPM is designed to handle FF function consisting of a gain, a first order lead and lag term plus dead-time, typical functionality that can be easily configured in most DCS systems.

ABB. All rights reserved

In addition, in order to increase the robustness, it is possible to design an additional first order filter defined by a gain ("**GainX**") and constant time ("**τ")**. Robustness filter parameters can be set by typing the requested values in the relevant text field or by moving two sliders.

For **Manual** FF tuning the user is fully responsible for selecting all the parameters of the compensator. For **Automatic** FF tuning LPM calculates all FF compensator parameters, while user must define the Robustness filter.
When clicking the Calculate button, the results are displayed in the "Results" frame and the simulation plot is also updated. These trends are simulations of the behavior of the loop, tuned with the newly calculated parameters, against the typical test-situation.
The chart displays the PV (upper chart) and the CO (lower chart) response to a Feedforward **disturbance** step, i.e. a step in the disturbance variable used for the feedforward, followed by the response to a step in the setpoint.

The size of the disturbance step applied in the simulation is shown below the Model selection (**Dist. Size**). You can alter the size from the provided default, and repeat the simulation, if you like.

There are four curves presented each time the Calculate button is pressed.

- Open loop disturbance response without feedforward.
- Open loop disturbance response with feedforward active.
- Closed loop responses without feedforward.
- Closed loop responses with feedforward active.

If the feedforward is ideal, it is very likely that the simulated responses with feedforward are so small that they are not visible at all. If the dead-times include fractions of the sample time, the responses may still be visible, also with the automatically calculated feedforward.

Below the charts there is the Frame "**Unit FF disturbance step**" containing numerical evaluation of the step responses with FF compensation active. In the frame three parameters are reported:

- CO oveshoot
- Max. deviation
- IAE

For the definition of these performance figures see paragraph 5.4.1.

5.4.4 FF Evaluation Tab

The FF Evaluation Tab (Figure 97 - FF Evaluation Tab) is devoted to investigate the effects on sensitivity function (see paragraph 5.4.2 for definition), when FF compensation is applied.

The graph shows the ability of the controller to reject disturbances. The 'disturbance passthrough' magnitude is plotted versus frequency on logarithmic scales. There are three curves:

- Closed Loop Disturbance Rejection with Feedforward Inactive
- Open loop disturbance rejection with Feedforward active
- Closed loop disturbance rejection with Feedforward active

ABB. All rights reserved

In the feedforward cases, the curve may be so low that it falls outside the diagram. (Log of 0 is minus infinity.)

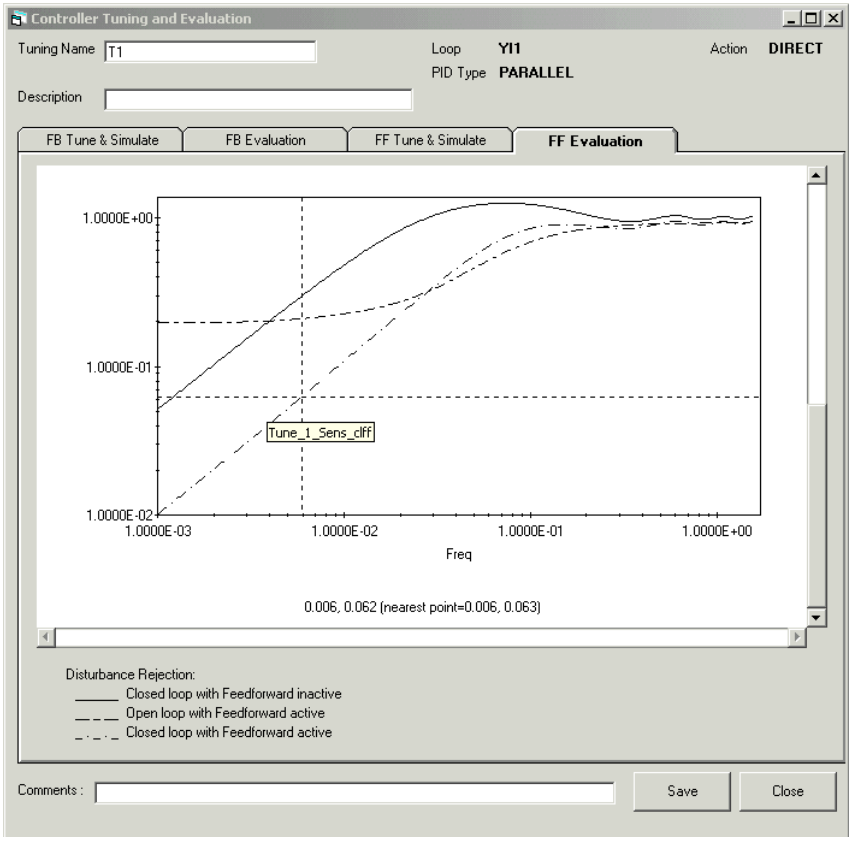


Figure 97 - FF Evaluation Tab

Each time the **Calculate** button in "**FF Tune & Simulate**" form is clicked, the new results are plotted. When the tuning session is over, just click on the Save button (to save the tuning parameters) or the Close button (to quit the session without saving) at the bottom of the form. A dialog similar to the one in Figure 98 - Save Tuning Constants will appear to check if the tuning parameters should be saved in the LOG files with a timestamp for future reference.

Figure 98 - Save Tuning Constants

ABB. All rights reserved

5.5 *Tuning Logs*

In order to access a Tuning Log, the user should double click on the Log icon of the desired loop or right click on the Log icon and select the View command (Figure 99 - View Log File).

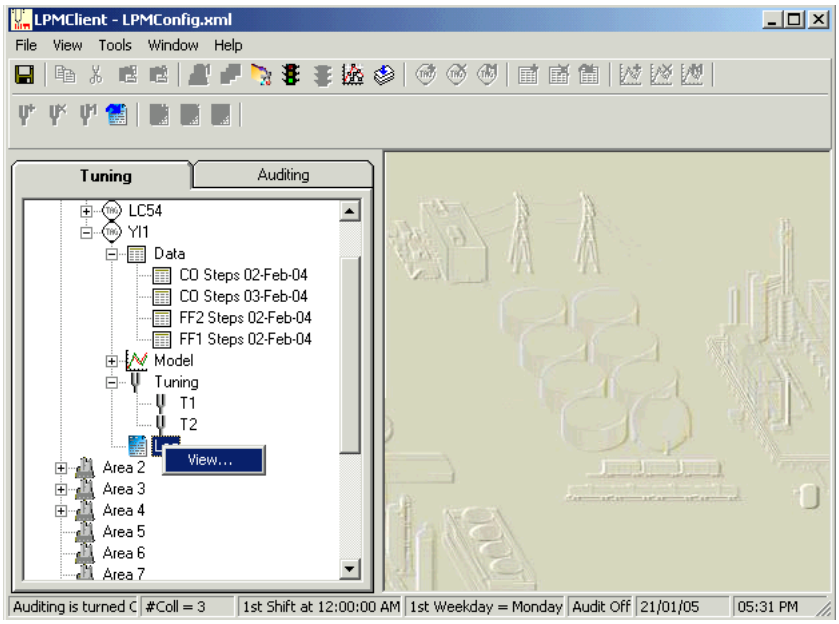


Figure 99 - View Log File

This action will open a form displaying all of the tuning actions, which had been previously saved to the log by the user. The information includes each saved set of tuning parameters together with a date and timestamp and information about which model was used for the tuning activity. If feedforward signals or cascade configuration are present, the relative information is appended soon below the Feedback section (see Figure 100 – Tuning Log).
This information provides a “loop tuning history” record for future reference and comparison.

ABB. All rights reserved

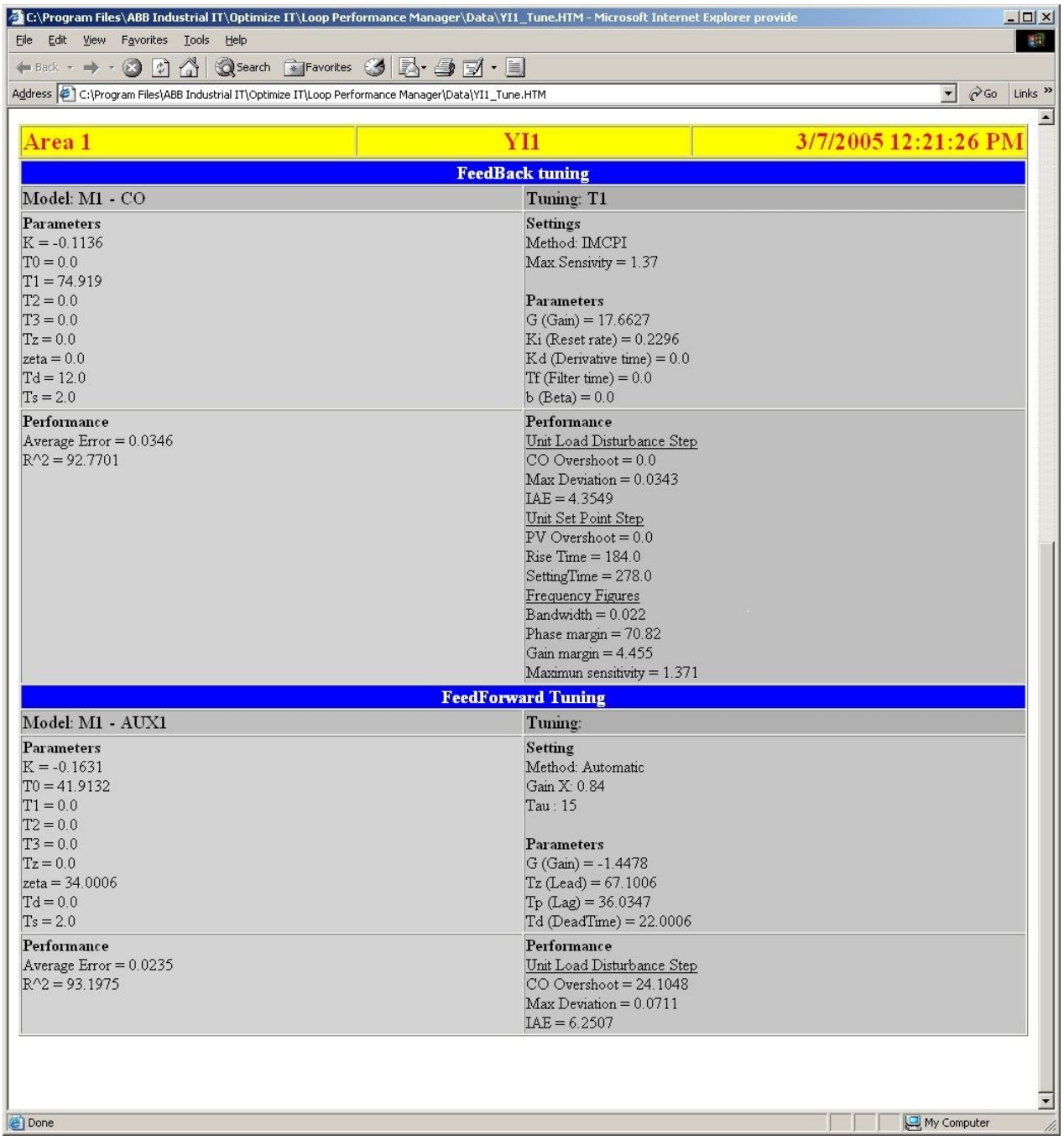


Figure 100 – Tuning Log

5.6 *Cascade Control Loops*

Figure 101 - Block diagram of cascade control scheme displays a block diagram of a cascade control scheme. There are two PID controllers, one for the inner loop and one for the outer loop. The control output of the outer loop PID controller is the setpoint of the inner loop PID controller. The block

ABB. All rights reserved

diagram also includes disturbance inputs and models for the inner loop and outer loop dynamics (Gp Inner and Gp Outer, respectively).

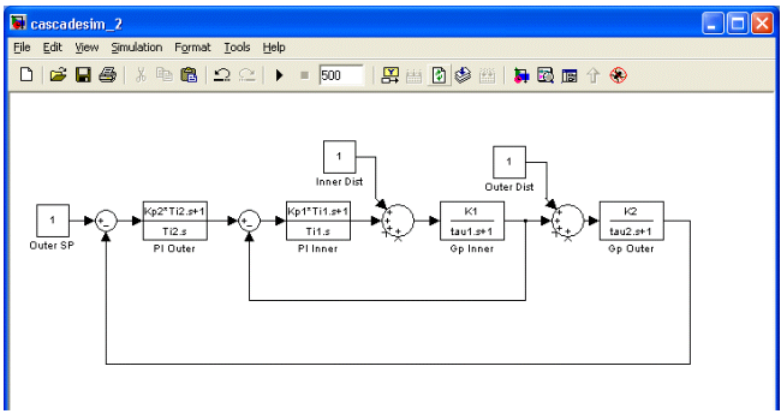


Figure 101 - Block diagram of cascade control scheme

In LPM, cascade controllers can be defined in one of two ways:

(iii) inner loop and outer loop are handled the same as any standard loop, or
(iv) outer loop is handled in a special manner by associating it with an inner loop.

In method (i), the user configures and tunes the inner loop and outer loop separately. The inner loop is tuned first, then the outer loop. The inner loop and outer loop are each treated as a standard PID control loop. When collecting data for the outer loop, the inner loop PID controller should be in AUTO mode and the outer loop model is developed from its control output (CO) signal to the outer loop process value. The tuning for the outer loop also proceeds in a standard manner as for any other PID control loop.

For method (ii), the user first defines the configuration and tuning of the inner loop in the same manner as for method (i) (i.e. same as a standard control loop), but then the outer loop is configured and tuned in a special manner.

In particular, the user should configure the outer loop as a Cascade Master (Outer) loop of a cascade control scheme and identify the name of the inner loop. Figure 102 - Configuration form for Cascade Master / Outer loop shows the loop configuration form for loop “LC54”.

ABB. All rights reserved

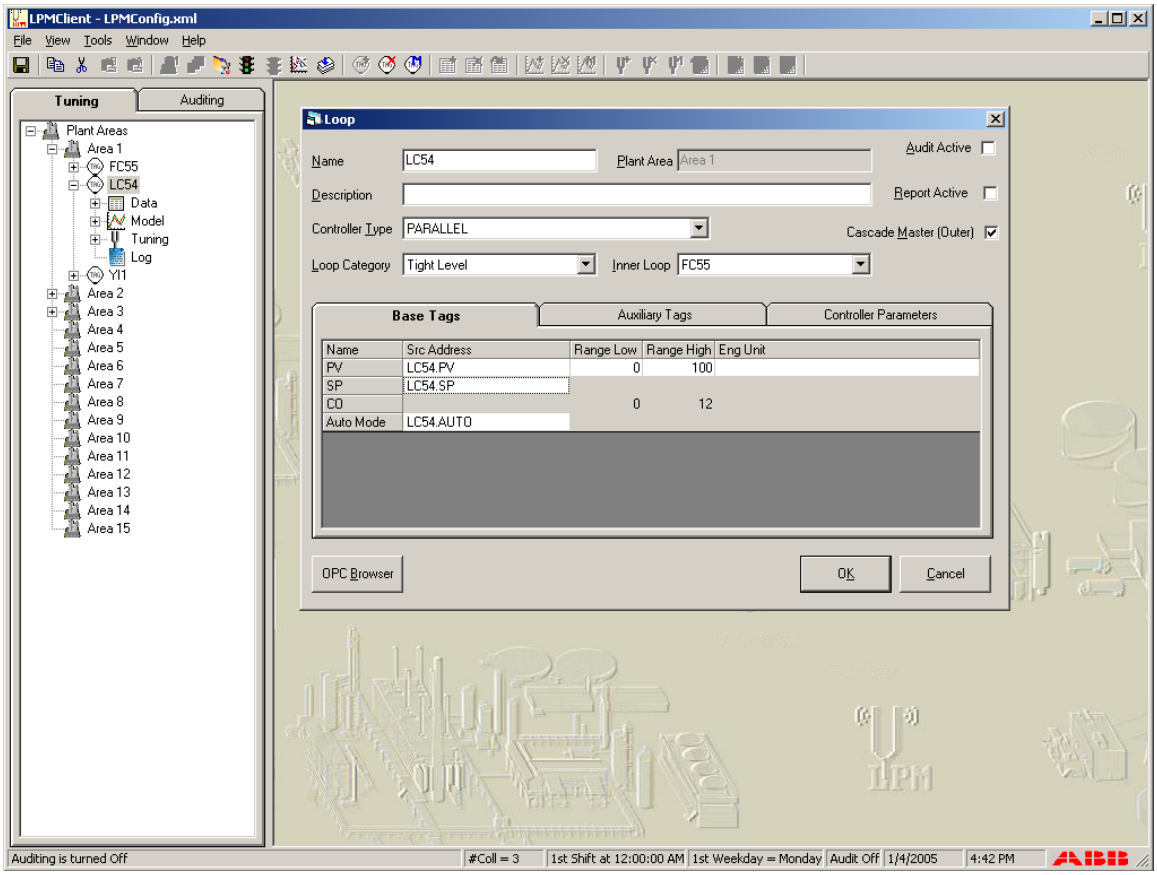


Figure 102 - Configuration form for Cascade Master / Outer loop

The user should click the checkbox labeled “Cascade Master (Outer)” to define it as an outer loop of a cascade control scheme. The user should then select the desired inner loop from the Inner Loop combo box, which lists all of the configured loops. Note that the Inner Loop combo box only appears if the Cascade Master check box is selected. Click on the OK button to save the loop configuration.

Note that the Src (Source) Address for the CO of the outer loop is not editable on the Base Tags tab of the loop configuration form. This Source address is automatically defined to be the same as the Source Address for the PV of the inner loop (to obtain the proper data for building the outer loop model).

When a loop is configured as a cascade master / outer loop, any new modeling and tuning objects will be treated differently from those defined for a standard loop. For example, in a modeling object of an outer loop, it is assumed that the model is defined between **the inner loop PV and the outer loop PV** (i.e. corresponding to Gp Outer) instead of between the outer loop CO and the outer loop PV. The outer loop model should be created as follows:

- Create a new data collection object for the outer loop. Start the data collection and ensure that the inner loop PV is changing sufficiently (to build model of Gp Outer). Save the dataset.
- Create a new modeling object. Select InnerPV as independent variable. Select the newly collected dataset, then choose appropriate modeling parameters and create a model. Save the

ABB. All rights reserved

new model. The new model will appear in the object tree with the extension "-CASC" (instead of "-CO") to indicate that it represents a cascade master / outer loop.

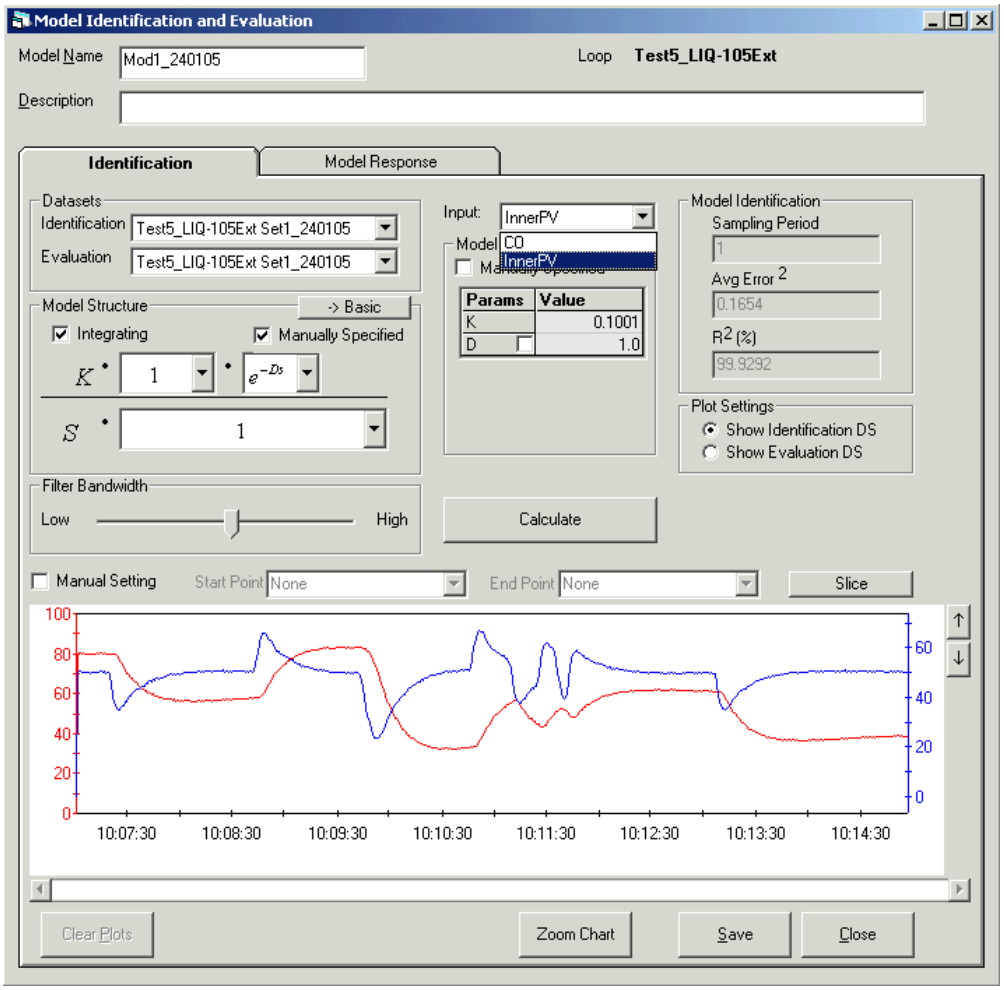


Figure 103 - Modeling form

Figure 104 - Tuning form for Cascade Master / Outer loop displays the tuning form for a cascade master / outer loop. Note that the tuning form displays the name of the inner loop as defined in the configuration form. The Tuning form also includes a combo box to select the desired inner loop tuning object. The inner loop tuning object defines the model and tuning parameters to be used for the inner loop when tuning and simulating the outer loop.

ABB. All rights reserved

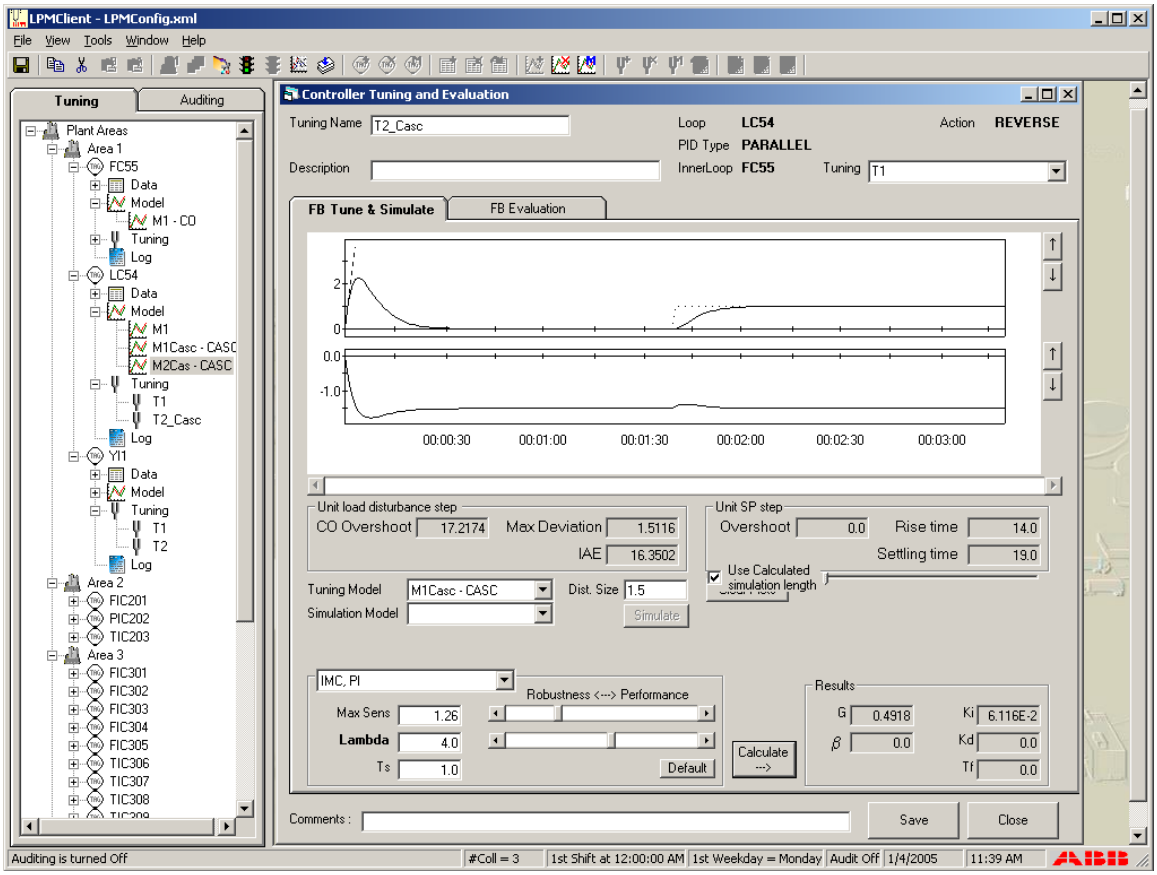


Figure 104 - Tuning form for Cascade Master / Outer loop

The user must also define the tuning method and tuning specifications (e.g. IMC PI, Lambda = 4.0) for the outer loop. When clicking on the Calculate button, the PID tuning parameters are displayed in the Results frame. Also, a chart displays simulation results, which include the Outer Loop CO (= Inner Loop SP), Outer Loop PV, and Outer Loop SP. The first half of the simulation corresponds to a disturbance at the process output of the inner loop. Note that if the outer loop is not configured as a cascade master, the simulation corresponds to a disturbance at the control output (of outer loop), which is typically the setpoint of the inner loop.

The FB Evaluation form displays the sensitivity function for the outer loop.

ABB. All rights reserved

6. Auditing

6.1 *Introduction*

The loop auditing tool is used to evaluate control loop performance. The aim of this tool is to allow automatic, non-invasive monitoring of many PID loops simultaneously, highlighting poorly performing loops and suggesting reasons for their poor performance.

Loop auditing includes the process of collecting, assessing and diagnosing data, and generating reports. Loops are classified into plant areas and loop categories. Each plant area may include a different data source (e.g. separate OPC server). For each loop category, the user may specify the frequency and duration for each data collection. The number of collections per day is also specified for all categories. Data is collected automatically for all loops in each category based on the specified collection parameters.

Data assessment includes the evaluation of the collected data to calculate performance indices, such as standard deviation, Harris index, oscillation index, etc. Diagnosis represents the historical evaluation of performance indices to identify equipment problems (e.g. valve stiction) or non-optimal tuning of control loops. A loop ranking report is automatically generated each week to highlight the worst performing loops and the problems identified by the diagnoses. Custom reports can also be configured and automatically created each week to display results for specific diagnoses, performance indices, plant areas, and loop categories.

6.1.1 Basic Nomenclature

Some definitions of basic terms related to loop auditing are provided in the following.

Data Collection Period Time interval between the start of each batch data collection. For example, if loop auditing is configured for 3 collections per day, then the data collection period is 8 hours. Data collection periods follow consecutively, e.g. if the first data collection period is 2003-12-01 00:00:00 to 2003-12-01 08:00:00, then the next data collection period is 2003-12-01 00:08:00 to 2003-12-01 16:00:00.

Batch data collection There is one batch data collection for each loop within each active loop category during a data collection period. The length of time for the batch data collection is called the duration and the time between each data sample is the sampling period (configured for each loop category).

Continuous data collection There is one continuous data collection for each loop within each active loop category during a data collection period. The length of time for the continuous data collection is equal to the auditing period and the time between each data sample is the sampling period (set to one minute).

Diagnosis Involves a series of hypothesis tests applied to the performance indices. A diagnosis result is presented as the percentage of time that the diagnosis was true over a time horizon (known as the diagnosis horizon).

ABB. All rights reserved

Diagnosis Horizon A period of time for which a diagnosis is based upon. Every report includes results from two diagnosis horizons: one week and one month.

Reporting Period Time period between automatic report generations (set to one week).

Loop Ranking Report Automatically generated Excel report that displays the overall loop performance index of each control loop, intended to provide a ranking of the loops The report highlights the worst performing loops and includes problem diagnoses for each of these loops. The report name consists of "Perf_Overview" followed by the date, e.g. "Perf_Overview 2003_November_20".

Custom Report A report configured by the user to include specific plant areas, loop categories, performance indices, and diagnoses. The report name consists of a user–specified prefix and the selected date and time; for example, "Loop Audit 2003_November_20".

6.1.2 Summary

There are two types of data collections for loop auditing:

- Batch monitoring
- Continuous monitoring

The batch monitoring data collections are organized by loop categories and are typically performed at higher sampling rates and shorter durations than the continuous monitoring collections. After a batch monitoring data collection is complete, several performance indices are calculated from this data and stored for future retrieval. These performance indices are then used in the diagnoses to detect problems, which can be highlighted in reports.

The continuous monitoring data collections are typically performed at a sampling rate of once per minute, but the collections continue indefinitely. Additional performance indices are calculated from the continuous monitoring data collections. These performance indices are calculated once for each auditing period, similar to the batch monitoring performance indices.

Figure 105 - Data For a Process shows the data (the black curve) for a process. To evaluate loop performance, you don't have to use all the process data. Instead, you collect batches of process data over several time periods. There are 3 gray-marked batch data collections in the diagram. The length of each collection is called the duration. In the diagram the duration is 1 hour. The interval between the start of each data collection is called the data collection period. It is 12 hours in the diagram.

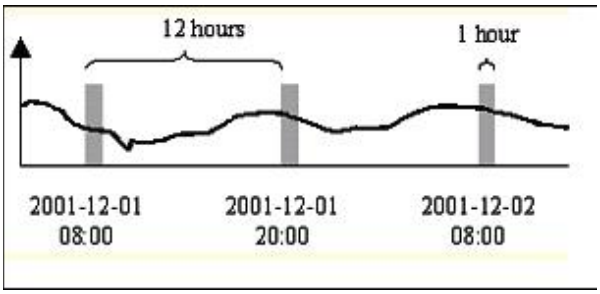


Figure 105 - Data For a Process

ABB. All rights reserved

Each data collection period is denoted by its starting date and time. The first collection period in the diagram starts at 08:00 on December 1, 2001 and the second period starts at 20:00 on December 1, 2001. The actual collection of data is only for 1 hour. The first collection starts at 08:00 and ends at 09:00 on December 1, 2001.

The number of auditing periods per day is a configurable parameter (between 1 and 24) that defines the number of data collections per day and the number of times per day that performance indices are calculated (since the indices are calculated once for each data collection).

Loop diagnoses and reports are created once per week based on the historical evaluation of performance indices. A diagnosis result is presented as the percentage of time that the diagnosis was true over a time horizon. Each report indicates diagnosis results for two time horizons:

- One week
- One month.

Therefore all data collections within the last week contribute to the one week diagnosis results and all collections within the last month contribute to the one month diagnosis results.
A report may also be created on demand using the last week and last month of data relative to the current time.

MS-Excel® files ***If the server does not start automatically after rebooting, then start the LPM service manually. This is done by first selecting Manage from the right click menu of the My Computer icon. Then click on Services and Applications to expand the list and click on Services. Then, select the LPM Auditing Server from the list and select Start from the right click menu to start this service*** {Figure 106 - Launching LPM Server service (1)}**.** ***When the service starts you should see both lpmsrv.exe and LPMServerExe.exe running as tasks in the Windows Task Manager. Then, click Properties and set the Startup Type option to "Automatic" to have the LPM Auditing Server start automatically in the future*** {Figure 107 - Launching LPM Server service (2)}**.**

ABB. All rights reserved

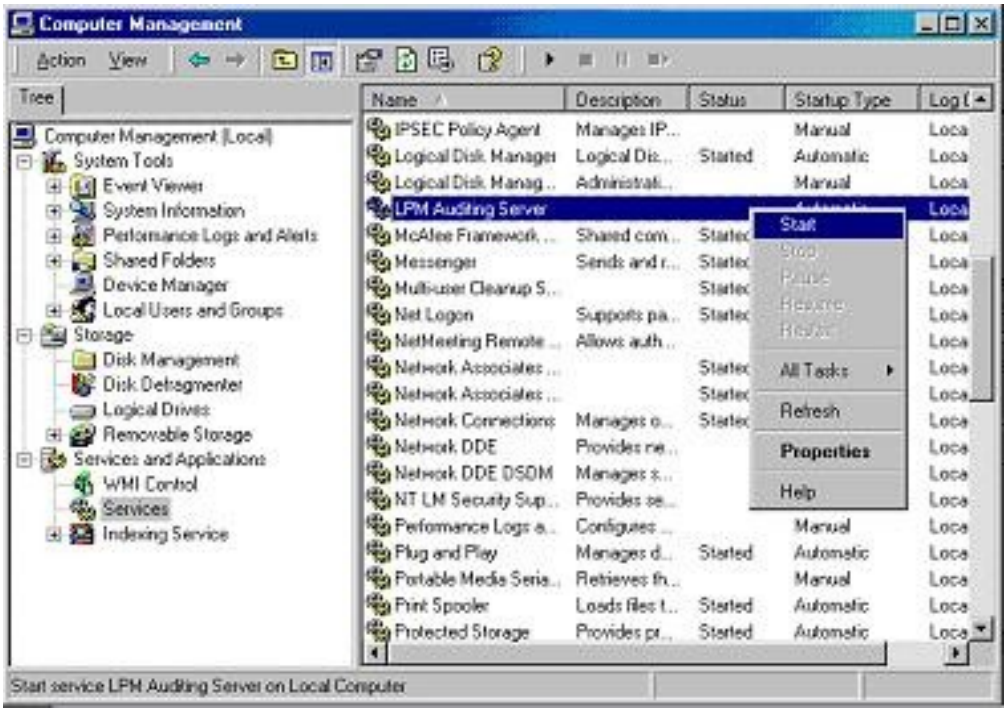


Figure 106 - Launching LPM Server service (1)

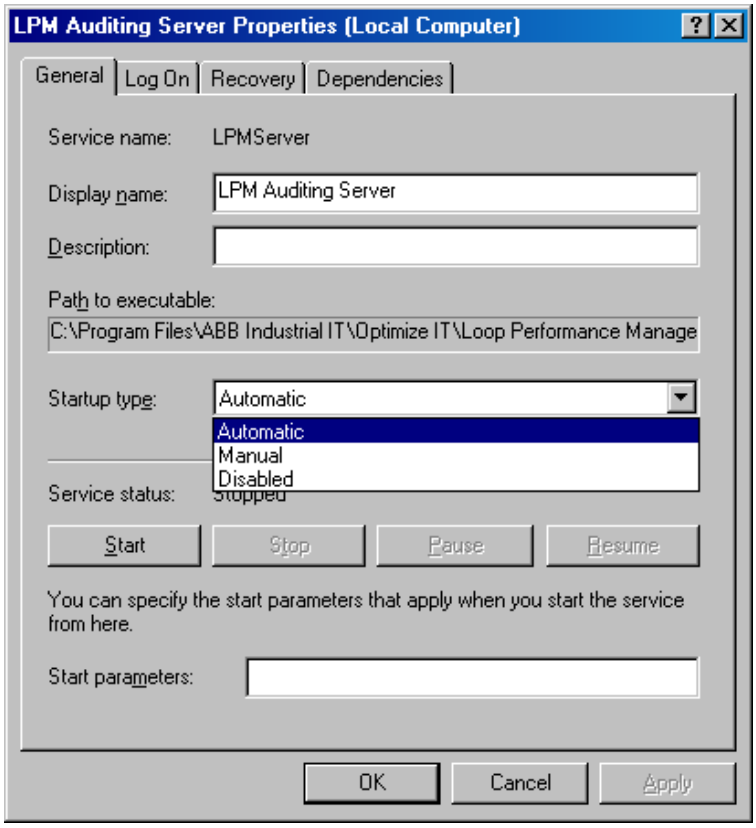


Figure 107 - Launching LPM Server service (2)

ABB. All rights reserved

6.2 *Configuration*

In order to properly configure an Auditing task, the user must select the right tab labeled as "Auditing" (see Figure 108 - Auditing Tree Structure).

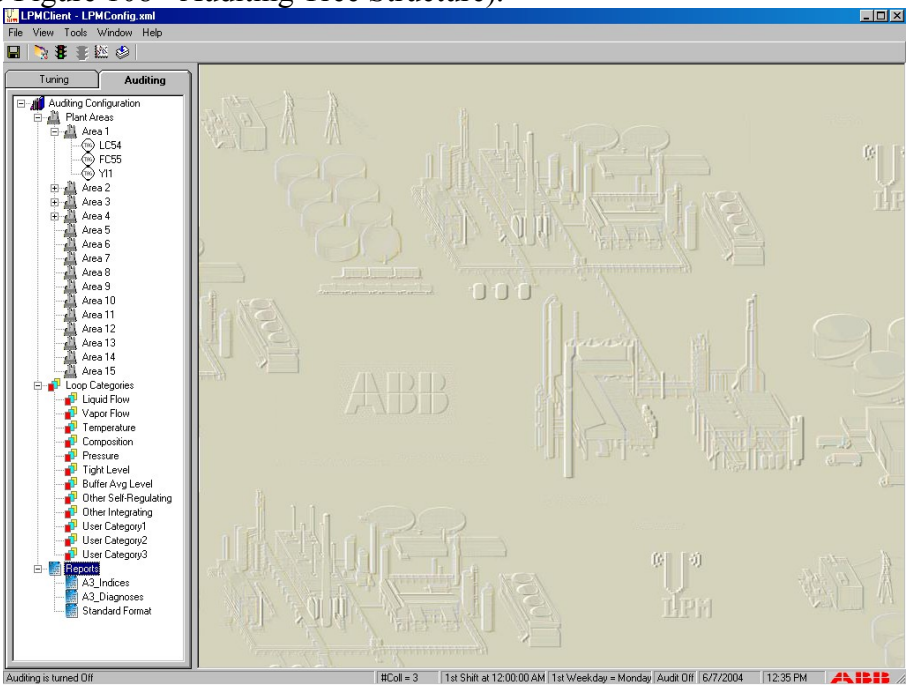


Figure 108 - Auditing Tree Structure

The Auditing tree shows three main object types:

- Plant Areas
- Loop Categories
- Reports

Each of these object types is described in the following sections to provide basic guidelines about how to navigate through the auditing tree.

First, a preliminary general configuration must be performed. In order to do so, the user must double click on the "Auditing Configuration" branch (or right click and select "Configure" as in Figure 109 - Auditing Configuration)

ABB. All rights reserved

Figure 109 - Auditing Configuration

This will launch a form similar to the one in Figure 110 - Auditing Shift Data Dialog.

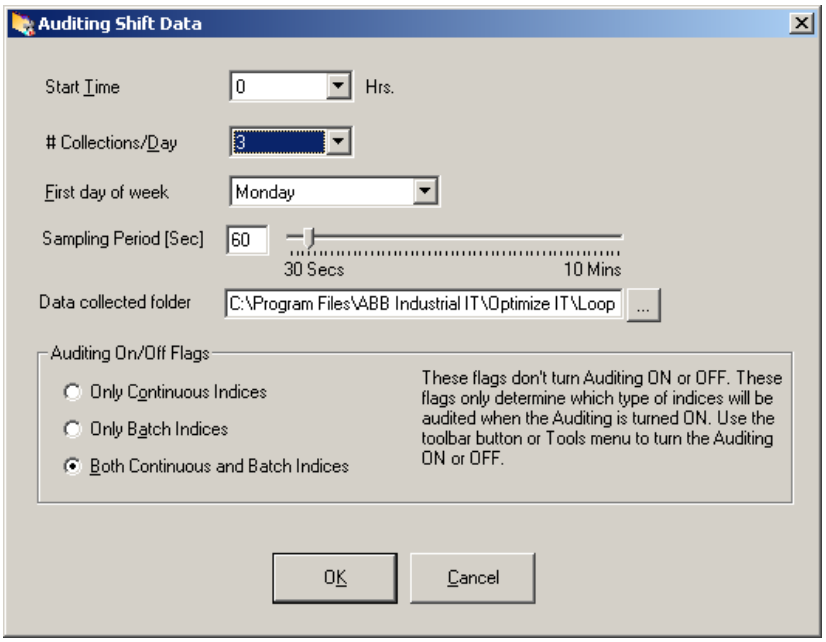

Auditing Shift Data
Start Time
0
Hrs.
Collections/Day
3
First day of week
Monday
Sampling Period [Sec]
60
30 Secs
10 Mins
Data collected folder
C:\Program Files\ABB Industrial IT\Optimize IT\Loop
Auditing On/Off Flags
Only Continuous Indices
Only Batch Indices
Both Continuous and Batch Indices
These flags don't turn Auditing ON or OFF. These flags only determine which type of indices will be audited when the Auditing is turned ON. Use the toolbar button or Tools menu to turn the Auditing ON or OFF.
OK
Cancel

Figure 110 - Auditing Shift Data Dialog

The user must select from the top-down menus the following three parameters:

- **Start Time**, i.e. time at which the auditing data collections are launched (note that only integer hours may be selected, e.g. "4" for 04:00);
- **# Collections/Day**, i.e. number of batch data collections per day (for each and every day). It is assumed that they will be equal-spaced along the day. So if 00:00 is selected as Start Time and 3 collections are set for each day, LPM will perform 3 auditing campaigns starting at 00:00; 08:00; and 16:00.
- **First day of week**, i.e. the day that defines the beginning of a new week. This information determines when the weekly report is issued.

Soon below the user can select the desired Sampling Period (choices are any value between 30 seconds and 10 minutes) simply acting on the related slider. Finally the user must specify the target directory where the data collected will be stored.

ABB. All rights reserved

6.2.1 Plant Areas

Plant Areas have been designed to allow the user to easily group the loops to be monitored and assessed in a logical, plant-wise manner. Typically (but not necessarily) a plant area is a process unit, or any other plant section to be singled out for the user. This will allow an ordered and easy archiving and retrieving of the loop information. Also, each plant area is allowed to have one data source (OPC server). Therefore, the user must assign loops to different plant areas if the data originates from different data sources.

Up to 15 Plant Areas are available in LPM. Each loop can be part of one and only one plant area. To configure a plant area, double click on the desired Area in the tree (or right click and select Configure).

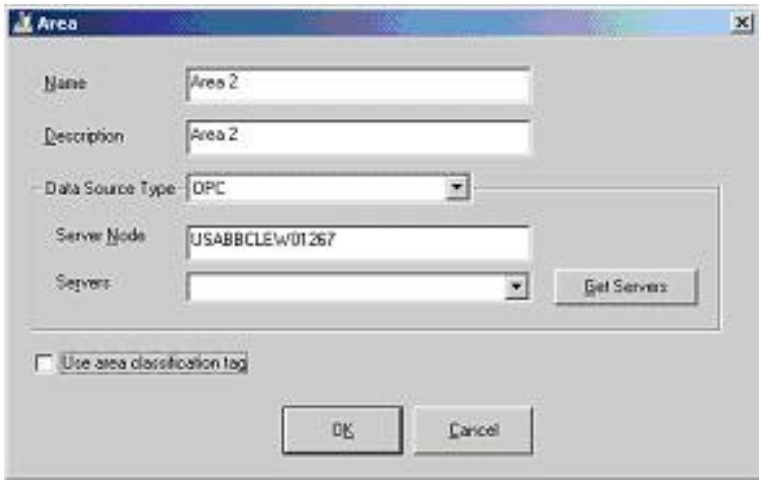


Figure 111 - Area Configuration Window

The user may enter a Name and Description for the plant area. The user must also select the data source type (OPC Server or Infi90/Symphony direct interface) and the server node and server name.

On the lower-left corner a checkbox is available to identify if the area under consideration should have an Area Classification Tag, or not. The concept and use of the Area Classification Tags will be described in § 6.4.2

6.2.2 Loops

In order to assign a loop into an Area, the user must double click (or right click) on the desired "Area" item and select "Add Loop" as in Figure 112 - Add Loop Dialog.

ABB. All rights reserved

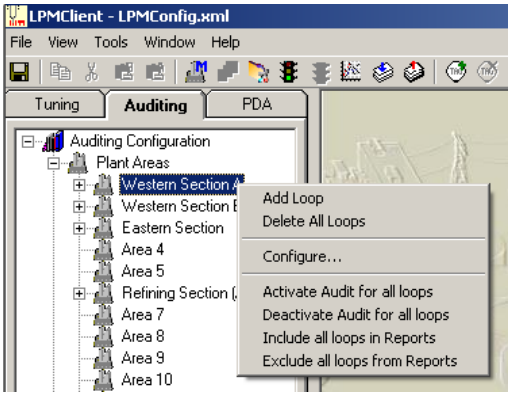


Figure 112 - Add Loop Dialog

A form similar to the one in Figure 113 - Add Loop Name will allow the user to enter the loop name.

Figure 113 - Add Loop Name

Figure 112 - Add Loop Dialog shows that right clicking on the Area icon it is also possible to:

- provide Area configuration;
- activate/deactivate auditing for all loops;
- include/exclude all loops in/from reports.

The above commands are designed to quickly change and update basic auditing task configuration.

Once the loop has been created it must be properly configured. In order to do so, double click on the loop name (or right click and select “Configure”) as in Figure 114 - Configure Loop.

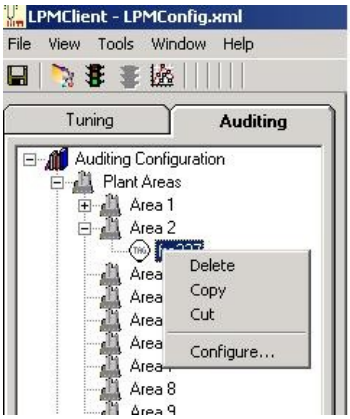


Figure 114 - Configure Loop

ABB. All rights reserved

The user will access a form structure similar to the one shown in Figure 115 - Loop Configuration Dialog: Base Tags. Each loop should be included in one of the categories listed in the drop-down menu shown on the left, while the drop-down menu on the right

Figure 115 - Loop Configuration Dialog: Base Tags

The top section is devoted to general information for the Loop (name, description), which may be edited. Two drop-down menus allow the user to select the Loop category and the PID Controller Type, as described in Section 4 for the Tuning setting.

Three checkmarks on the top-right corner allows the user to specify:

a) if the selected loop is to be included in the Auditing data collections (`Audit Active`). A new loop is active by default. The user can deactivate a loop from auditing data collections if it is out of service, etc.
b) if the selected loop performances must be included into the periodic reports (`Report Active`). A new loop is not included into the reports by default.
c) If the selected loop is the Master (outer, external) loop in a Cascade configuration (`Cascade Master (Outer)`). A new loop is not considered to be an Outer loop in a Cascade configuration by default.

The bottom portion of the form consists of three tabs:

- Base Tags
- Auxiliary Tags
- Controller Parameters

The Base Tags tab allows the user to enter basic information (ranges, engineering units, data source tagname) for the loop PV, SP, and CO and the Auto Mode (Figure 115 - Loop Configuration Dialog: Base Tags).

ABB. All rights reserved

If the Cascade Master checkbox is selected, the tab will automatically change, as described in Figure 116 – Loop Configuration for Cascade Master. The CO cell will be grayed out and made unavailable, while a new tag called "`Inner PV`" will be added to the Base Tags tab. The Src Address for this new tag is automatically set = Src Address of inner loop PV and not be editable by user. The Src Address of outer loop CO should be automatically set = Src Address of inner loop SP and is not editable.

Figure 116 – Loop Configuration for Cascade Master

The Auxiliary Tag tab is devoted to insert the needed information about one or more auxiliary signals (e.g. Feedforward variables) affecting the loop as shown in Figure 117 - Loop Configuration: Auxiliary tags. Note that the information requested in this form is not needed for auditing.

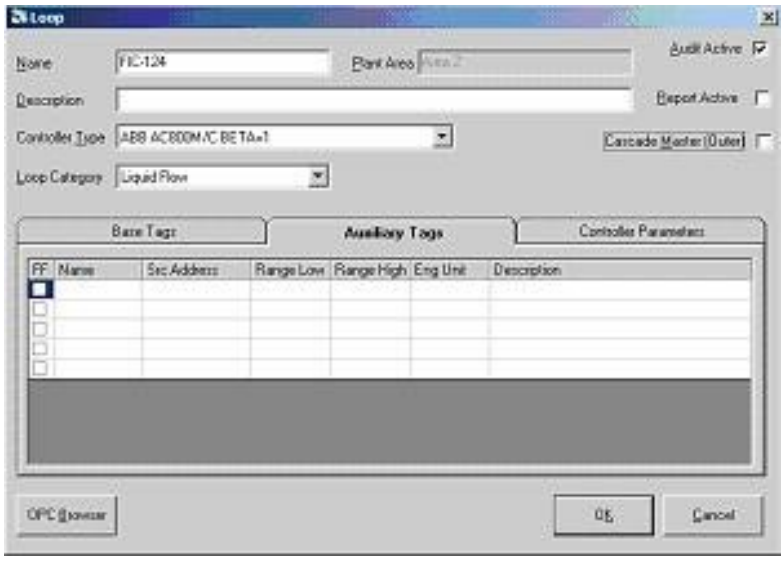


Figure 117 - Loop Configuration: Auxiliary tags

ABB. All rights reserved

Finally the Controller Parameters tab is devoted to the introduction of controller constants (Figure 118- Loop Configuration: Controller Constants). Please note that this form is controller sensitive, i.e. the list changes in order to match the given controller specifications. Note that the information requested in this form is not needed for auditing.

Figure 118- Loop Configuration: Controller Constants

At the bottom-left corner the "OPC Browser" button allows to access the form shown in Figure 119 - OPC Browser Dialog.

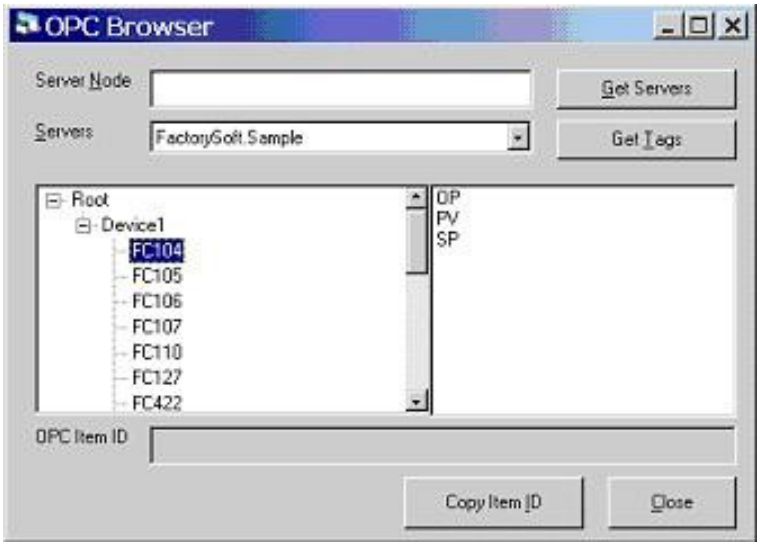


Figure 119 - OPC Browser Dialog

This form allows the user to select the OPC server (through the "`Get Servers`" button) and the desired tag (through the "`Get Tags`" button) and then establish the desired connection with the DCS.

As described in 5.1.2.2, an alternative quicker way to configure loops is via the Bulk Data Configuration Utility.

ABB. All rights reserved

6.2.3 Loop Categories

The Loop Categories have been designed to allow the user to properly group the loops to be monitored and assessed. The categories allow for easy archiving, retrieving and comparison of the loop information and also provide a natural organization of loop types for calculating performance indices and diagnoses.

Up to 12 loop categories are available for each auditing task: 9 are pre-defined, covering the plant typical loop categories, while 3 additional ones have been left available for customer configuration:

- Liquid Flow
- Vapor Flow
- Temperature
- Composition
- Pressure
- Tight level
- Buffer Avg Level
- Other Self-regulating
- Other Integrating
- User Category 1
- User Category 2
- User Category 3

Each loop can be part of one and only one loop category.

Each loop category must be configured for auditing purposes. In order to do so the user must double click (or right click) on the desired category item and select “Configure” as in Figure 120 - Loop Category Configuration.

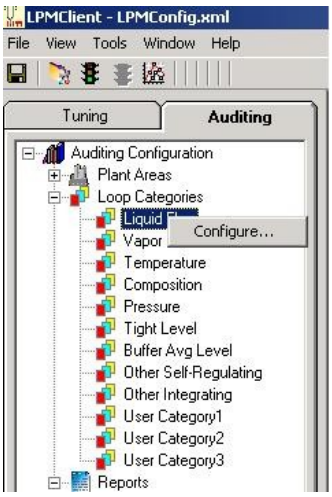


Figure 120 - Loop Category Configuration

This will make available the form shown in Figure 121 - Loop Category Configuration Window

ABB. All rights reserved

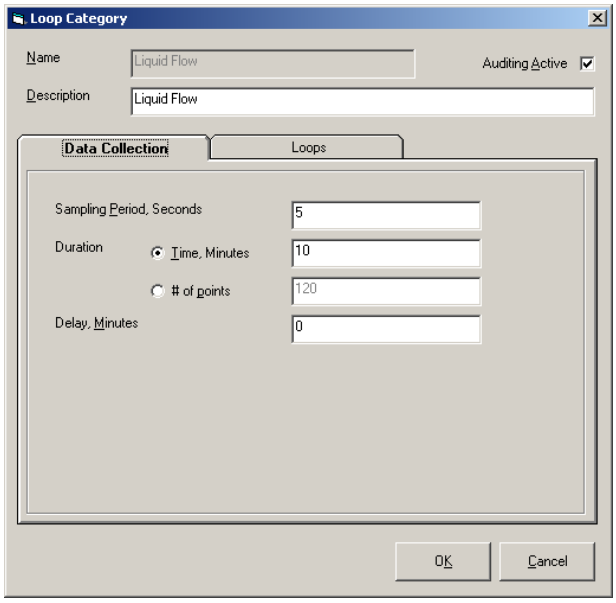


Figure 121 - Loop Category Configuration Window

The form is made up of two different tabs:

- **Data Collection** tab
- **Loops** tab.

On the Data Collection tab the user can introduce the category-specific data collection parameters, which are:

- Sampling Period (to be expressed in seconds),
- Duration (in minutes), and
- Delay (in minutes).

Sampling Period indicates the time period at which data is collected for each loop of the category during each batch data collection. The Duration parameter indicates the time period over which data will be collected for each loop of the category during each batch collection. As an alternative, the user may specify the number of data points to be collected for each batch collection instead of specifying the time duration. Given the number of points and the sampling period, the duration in minutes is automatically calculated and displayed. Similarly, given the duration in minutes and sampling period, the number of points is collected and displayed.

The Delay is also specified in minutes and indicates the period of time to wait (delay) the beginning of each batch data collection from the start of the collection period. For example, if the data collection period is to begin at 08:00:00 and the delay is 60 minutes, then the batch data collection begins at 09:00:00 for all loops in this loop category. The delay is primarily used to spread out the computational load when the loop auditing tasks are performed for a large number of loops.

Note that continuous data collections are performed automatically with the selected sampling period (in the range of 30 second to 10 minutes) for all loops in all loop categories.

ABB. All rights reserved

The Loops tab (Figure 122 - Loop Category Configuration, the Loops tab) displays all of the loops assigned to this category (as defined in the Loop Configuration form). The checkbox on the left of the Tagname identify tags active for auditing purposes and corresponds to the checkbox of the individual loop configuration as shown in Figure 115 - Loop Configuration Dialog: Base Tags.
For the auditing functionality to be functional, the Auditing Active checkbox should be selected.

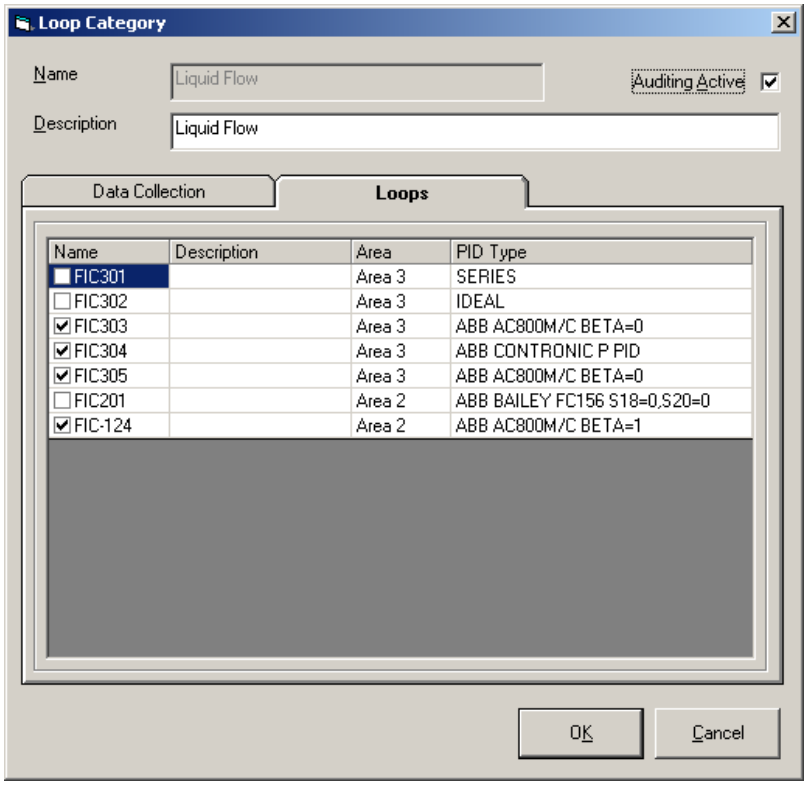


Figure 122 - Loop Category Configuration, the Loops tab

6.2.3.1 Batch Category/Shifts Résumé

Proper batch data collection scheduling is crucial to attain proper loop auditing performances. The right trade-off between short sampling time (so to catch important information) and reasonable data collection loading must be achieved. Although data collection for auditing purposes does not need the high frequency requested for model identification/loop tuning (loop auditing looking for slowly developing effects), it is important not to under sample and to properly differentiate by categories as described in 6.2.3 .
In order to help the user to quickly assess both the scheduling and the related load LPM makes available a devoted utility called Batch Category/Shifts Resume. This can be launched by right clicking on each category and then selecting the `Batch category/Shifts Resume` option, as shown in Figure 123 – Batch category shift configuration menu.

ABB. All rights reserved

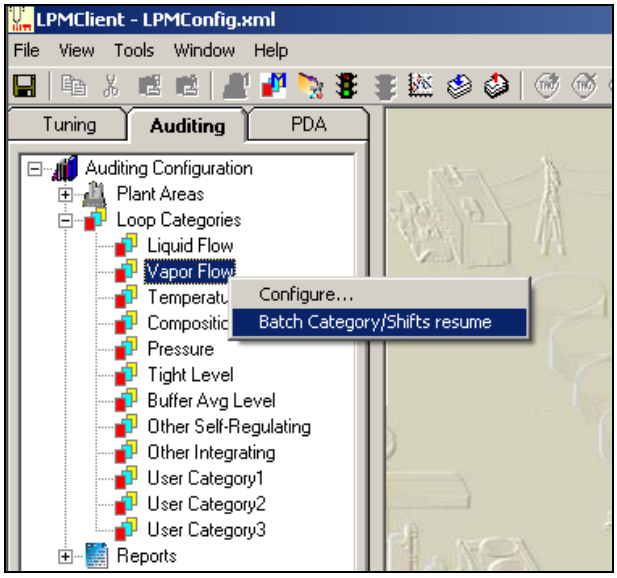


Figure 123 – Batch category shift configuration menu

A form like the one shown in Figure 124 - Batch Category/Shifts Resume, will pop-up showing:

1. in the upper part, how the data collection has been scheduled for each category;
2. in the lower part, the related load at each time.

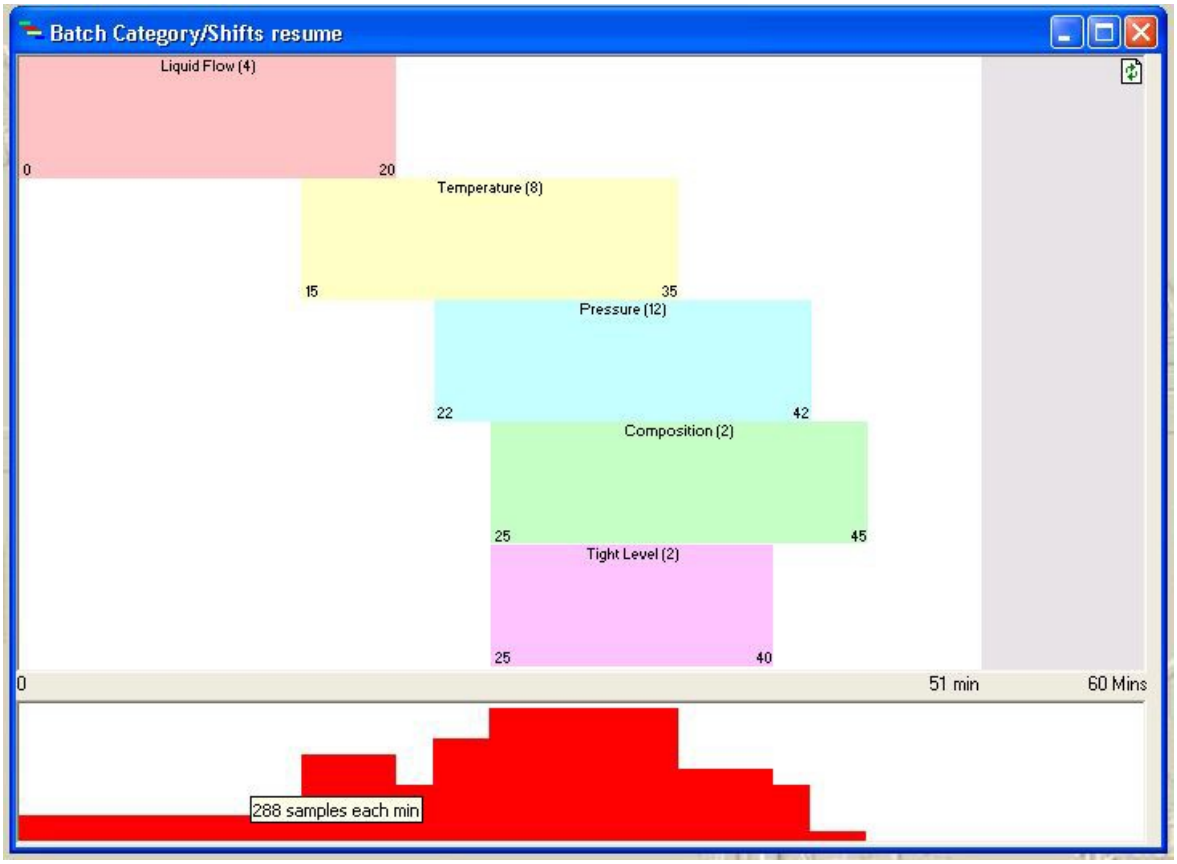


Figure 124 - Batch Category/Shifts Resume

ABB. All rights reserved

Note that – in the upper part - for each category (identified also by a color code) the form shows when the data collection starts and ends (in minutes after the start of the shift) and how many loops are configured for the given category. At the right end a grey band emphasizes the part of the time shift which should be left without data collection so to allow the sw to properly save and prepare for the following shift. Right clicking on each period a menu like the one shown in Figure 125 – Batch category shift setup is shown.
Through proper selection the user can set or remove a start time (for the entire shift), show or hide the lower section (Load) or pop-up the properties form (see Figure 122 - Loop Category Configuration, the Loops tab) for the given category.

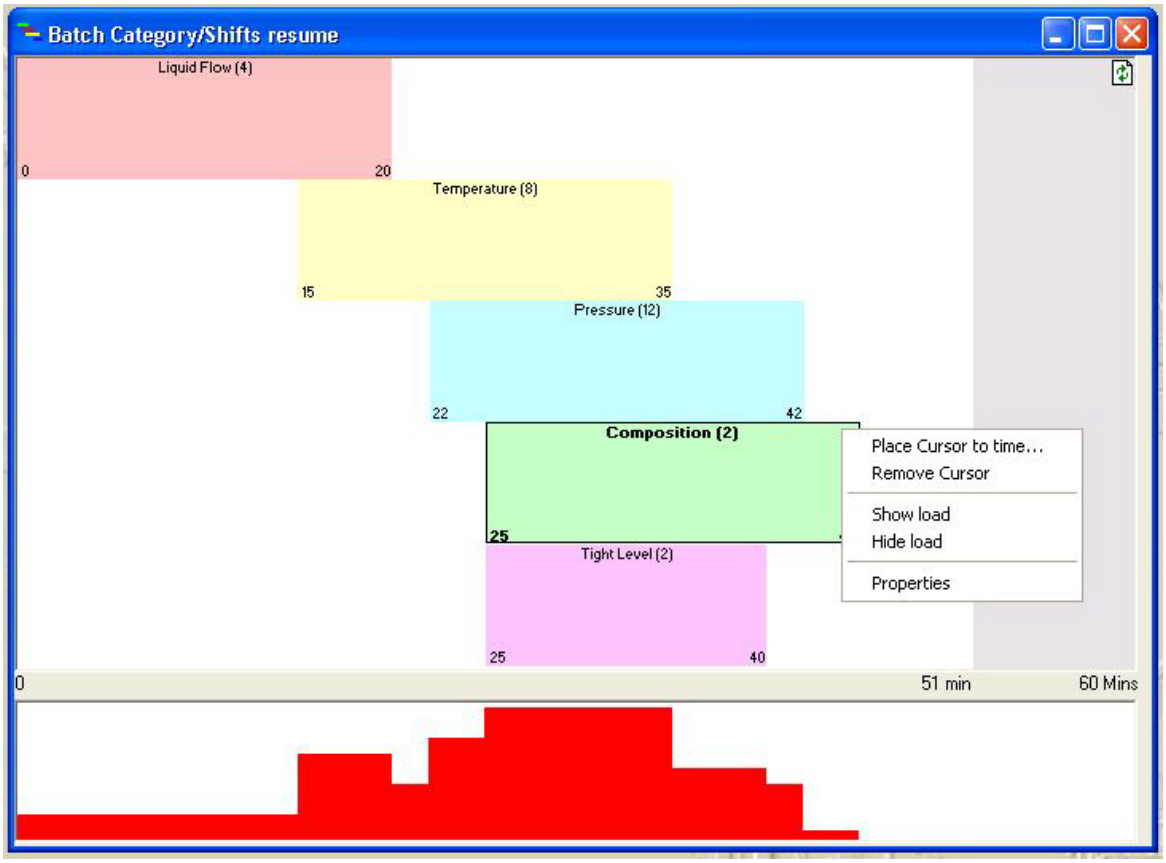


Figure 125 – Batch category shift setup

In the lower section the user can check the load on the acquisition channel, expressed in data per minute.
The user should check that these numbers are compatible with the DCS and data acquisition capabilities in order not to overload the system.

ABB. All rights reserved

6.3 *Assessing Loop Performances*

6.3.1 Performance Indices

Loop assessment includes the calculation, storage, and display of a number of performance indices to assist in the evaluation of the performance of control loops. LPM computes and manages the performance indices and parameters reported in Table 5.

Index	index class
mean loop error	Basic statistics
std loop error (norm)	Basic statistics
std loop error (eng.unit)	Basic statistics
mean PV	Basic statistics
std CO	Basic statistics
skewness control error	Basic statistics
kurtosis control error	Basic statistics
compression test (value)	Data validity
quantization test (value)	Data validity
outlier	Data validity
noise index	Loop modes
ratio CE / CO	Loop modes
automatic mode	Loop modes
saturation	Loop modes
cascade mode	Loop modes
ACF/pred.horizon ratio	Performance
setpoint crossing index	Performance
cascade tracking index	Performance
cascade travel ratio	Performance
Harris index (FCOR)	Performance
oscillation index CE	Oscillation
oscillation index SP	Oscillation
oscillation period	Oscillation
oscillation amplitude	Oscillation
oscillation severity	Oscillation
shut-off PV value	Valve
stiction (correlation)	Valve
stiction (histogram)	Valve
valve travel per hour	Valve
valve reversals per hour	Valve
valve sizing index (10/90 time)	Valve
nonlinearity measure	Nonlinearity
NLI	Nonlinearity
NGI	Nonlinearity
Oscillation symmetry	Nonlinearity
estimated time-delay	Property
master flag	Property
step flag	Property
model fit flag	Property
static gain	Property
prediction horizon	Housekeeping
loop category	Housekeeping
classification value (optional)	Housekeeping
user-defined index # 1	Special
user-defined index # 2	Special
user-defined index # 3	Special
Auto mode (logged)	Continuous
Control error value	Continuous
percentage of saturated (index?)	Continuous

Table 5 - List of Auditing Performance Indices

ABB. All rights reserved

As the color code clearly identifies the 51 indices are grouped in the following 11 categories:

1. ***Basic statistics***
2. ***Data validity***
3. ***Loop modes***
4. ***Performance***
5. ***Oscillation-related***
6. ***Valve-related***
7. ***Non-linearity***
8. ***Property***
9. ***Housekeeping***
10. ***Special***
11. ***Continuous***

While referring to Appendix 11 for a detailed description of each index, it is worth to be noted that the first 45 indices are calculated from the batch data collections and are used in the diagnoses. Data from the loop setpoint, process value, and control output are required to calculate these indices.
Indices 46, 47, 48 (belonging to the "Special" category) are actually void and earmarked as User-defined" in order to let the advanced user create its own indices, if needed.

Finally, indices 49 through 51 are calculated from the continuous data collections. Index 39 (% of time in auto mode) is calculated from the data collected for the auto mode tag. It is assumed that a data value of 0 indicates a non-Auto mode while any other value indicates Auto mode.

6.3.2 Diagnoses

Based on the combined results of the above-described indices, LPM computes 11 diagnostic hypotheses, listed in Table 6 - LPM Diagnostic Hypothesis and Pre-conditions.

The 7 Pre-conditions, listed at the bottom of Table 6 - LPM Diagnostic Hypothesis and Pre-conditions are actually used to determine the main, summarizing Hypothesis H1:

$$\mathbf{H_1 = P_1 \wedge P_2 \wedge .... \wedge P_7}$$

ABB. All rights reserved

Hypothesis	
acceptable overall performance	1
tuning problem	2
loop oscillatory	3
setpoint oscillatory	4
significant external disturbance	5
significant nonlinearity	6
valve stiction	7
valve leakage or zero error	8
valve size incorrect	9
excessive valve action	10
data unrealiable	11
Pre-conditions	
acceptable Harris index	1
acceptable setpoint crossings index	2
variability random	3
controller output within range	4
loop automatic	5
acceptable cascade tracking	6
acceptable response speed	7

Table 6 - LPM Diagnostic Hypothesis and Pre-conditions

The loop diagnoses are calculated once per week. The diagnoses are calculated for two horizons: one week and one month. Therefore, one set of diagnosis results are based on the last week of performance indices and the other set of results are based on the last month of performance indices. The diagnosis results are presented in reports as the percentage of time that the diagnosis was true during the diagnosis horizon.

6.3.3 Plotting Indices and Diagnoses

The LPM Auditing tool also allows the user to plot performance indices versus time for one or more loops. To create a data plot, the user must first click on the Data Plot icon or select Plot Indices from the Tools menu, as shown in Figure 126 _Plot Indices Selection.

ABB. All rights reserved

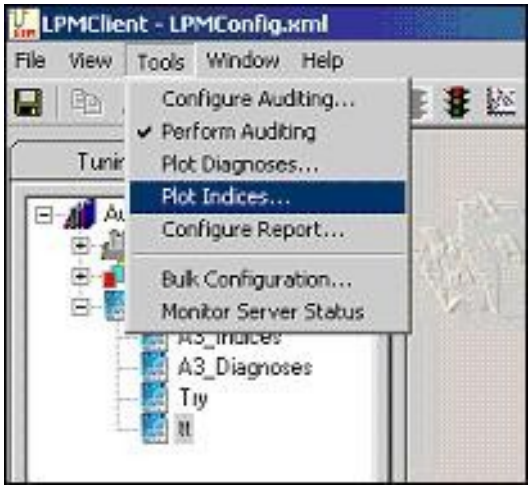


Figure 126 _Plot Indices Selection

A form will pop-up, allowing the user to select which loops (up to 8 per plot), which indices (up to 6) and which time period are desired. Figure 127 - Plot Indices: Append Month Mode shows one of the two possible plotting options (Append Month), useful when trending indices over multiple periods is requested.
Like all the other LPM plots and trends, multiple visual functions (zooming, trending set-ups, etc.) are available: Figure 128 - Stacked Plot Option shows, as an example, the same plots of Figure 127 - Plot Indices: Append Month Mode individually trended in a stacked chart.

ABB. All rights reserved

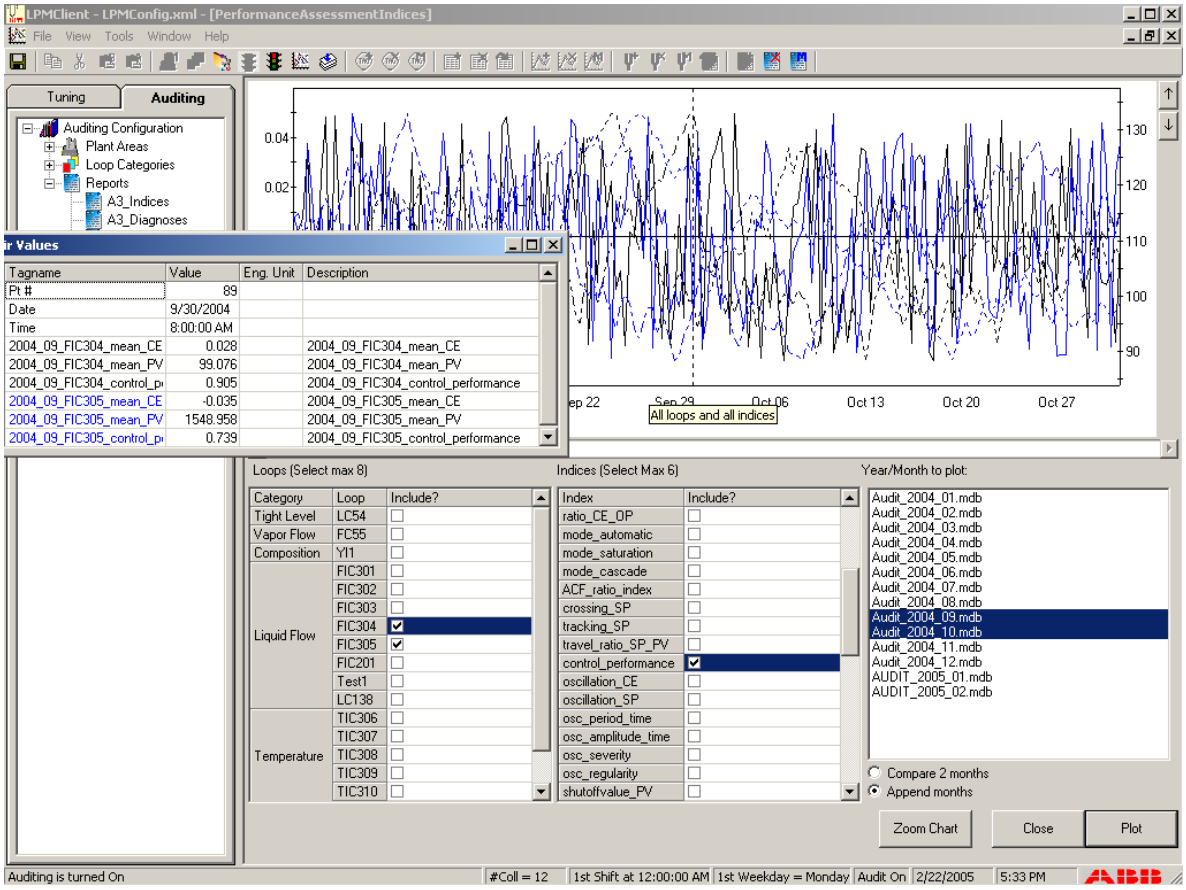


Figure 127 - Plot Indices: Append Month Mode

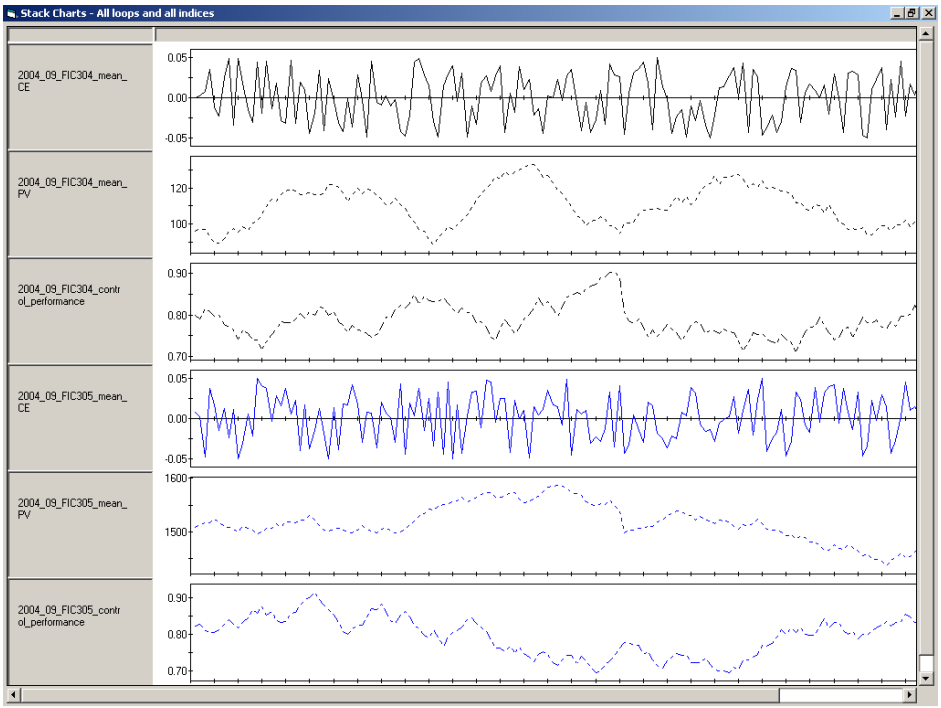


Figure 128 - Stacked Plot Option

ABB. All rights reserved

A second option ("`Compare Months`") is available, when the user wants to compare one or more indices of one or more loops over different time periods. Figure 129 –Compare Months shows the comparison of the control performance of loop FIC304 between the months of July and October 2004.

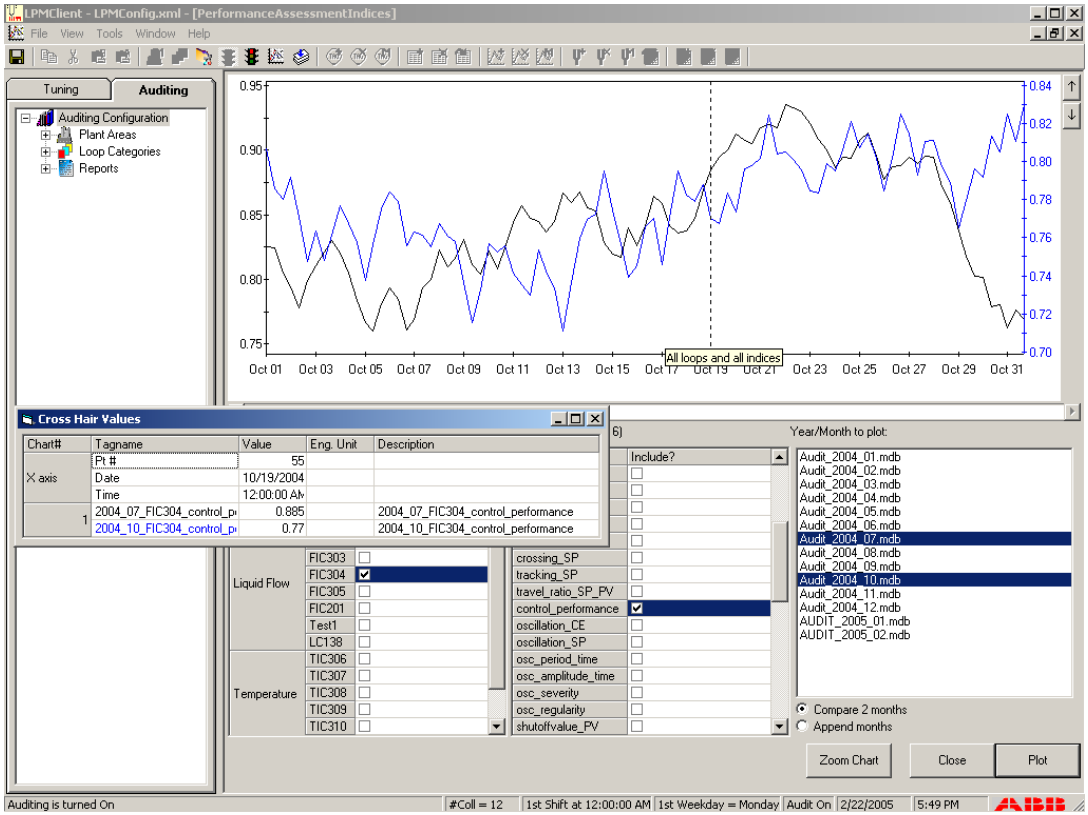


Figure 129 –Compare Months

A similar facility (accessible from the Tools menu, as well) is available also for plotting Diagnosis in a complete identical manner.

ABB. All rights reserved

6.4 *Reports*

LPM is designed to produce customisable reports, which are automatically saved and stored so to allow maintenance department to access them for regular assessments. Report generation can happen in two ways:

- Periodically (i.e. the software automatically generates reports each week and month)

Note: Customizable reports should be created & configured by user so that LPM will generate periodic reports automatically. Follow Report Configuration

- On Demand (i.e. the user can request the generation of an additional report every time there is a need or a special process condition)

A new loop ranking report is created automatically every week (at the start time defined in the auditing configuration form). The user may also configure custom reports by selecting specific plant areas, loop categories, performance indices, and diagnoses. Configured custom reports will also be created automatically every week.

Reports are created both as Excel spreadsheets and in HTML formats.
Note: To create HTML report automatically user need to check the "Save report also in HTML Format" under Tools->Options->Options Tab in LPM Client (as shown in figure below).

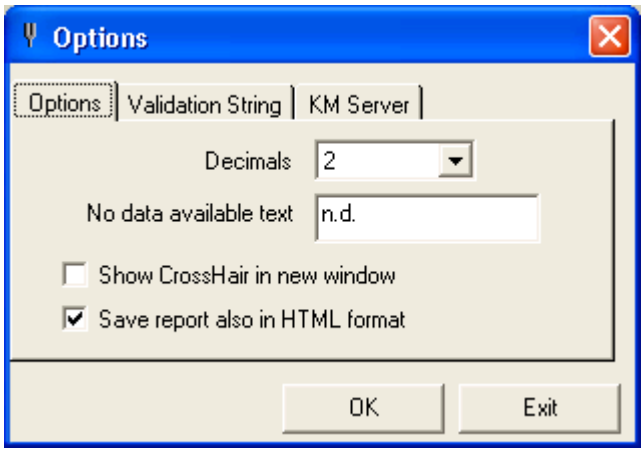


Figure 130 - Options

In particular, the report ranks loops within each category according to the results for diagnosis H1 “Acceptable Overall Performance”. The value of the acceptable overall performance diagnosis is the weighted average of the percentage scores for 8 hypothesis tests included in the overall performance ranking.

The report also provides a graphical plot of the performance for each loop category and for all categories. It is based on the following performance classification:

ABB. All rights reserved

- Excellent
- Good
- Fair
- Poor

The loop ranking report also identifies the “Poor” performing loops for each loop category. For each loop listed as poor, the report includes the name and overall score for the two worst tests of the 8 tests used in computing the overall performance (H1). This information makes it easy to identify the worst performing loops and indicates the diagnoses, so that problems can be corrected to improve loop performance.

6.4.1 Report Configuration

A custom report can be configured by clicking on the Reports branch of the tree and electing Add item (Figure 131 - Add report Dialog).

Figure 131 - Add report Dialog

A dialog allowing the insertion of the report name is then opened.
The Report Configuration form is made up of five different tabs:

- **Loop Categories** tab;
- **Plant Areas** tab;
- **Indices** tab;
- **Diagnoses** tab;
- **Plot** tab;

The `Loop Categories` tab (Figure 132 - Report Configuration: Loop Categories) allows the user to select the categories to include in the report.

ABB. All rights reserved

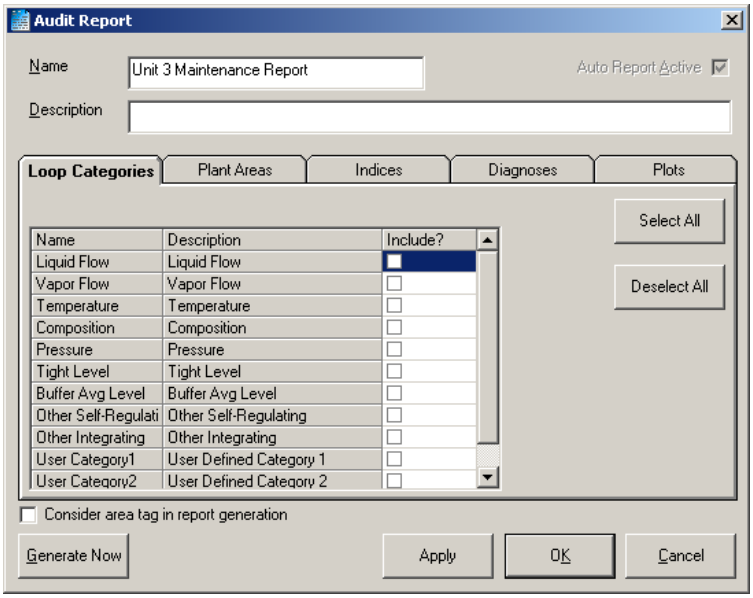


Figure 132 - Report Configuration: Loop Categories

In the same manner, the `Plant Areas` tab (Figure 133 - Report Configuration: Plant Areas) allows the user to select plant areas to include in the report.

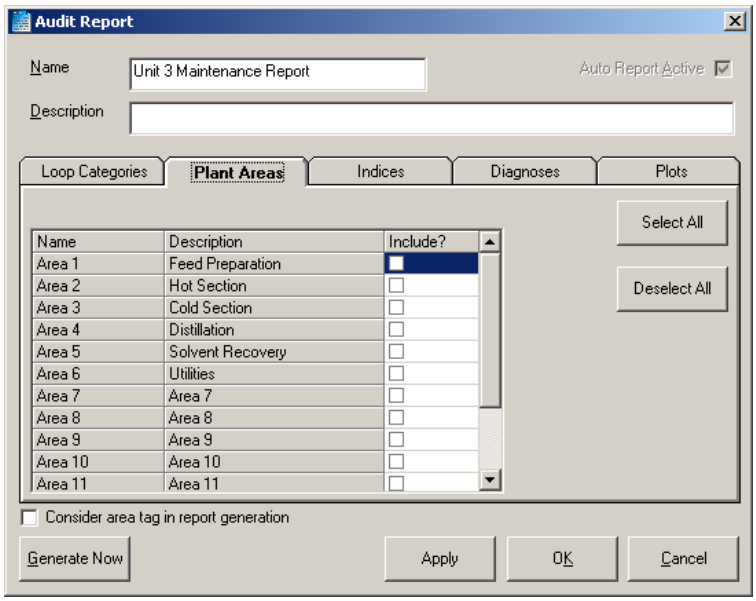


Figure 133 - Report Configuration: Plant Areas

The `Indices` tab (Figure 134 - Report Configuration: Indices) allows the user to select the performance indices to include in the report.

ABB. All rights reserved

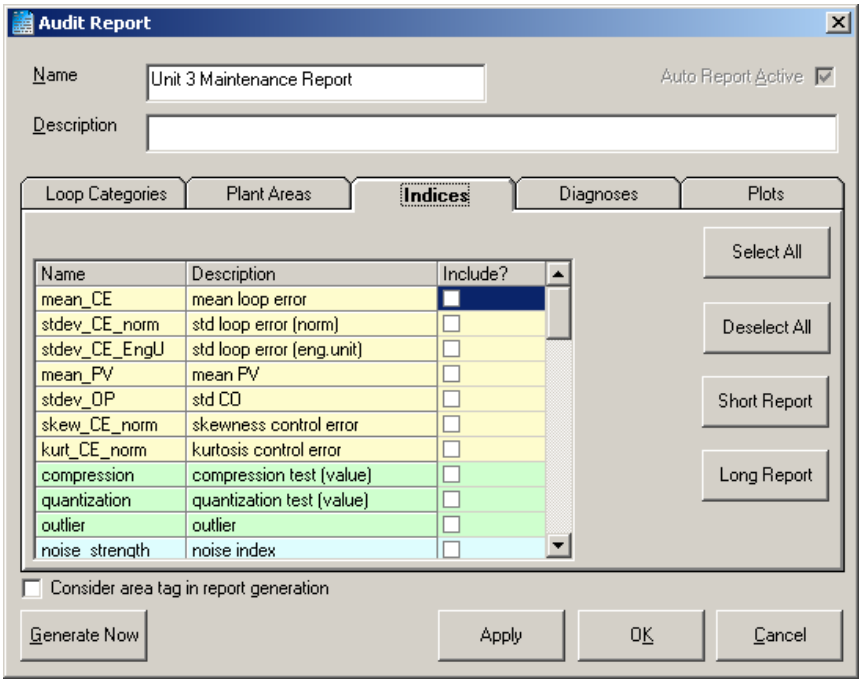


Figure 134 - Report Configuration: Indices

The `Diagnosis` tab (Figure 135 - Report Configuration: Diagnoses) allows the user to select the diagnoses to include in the report.

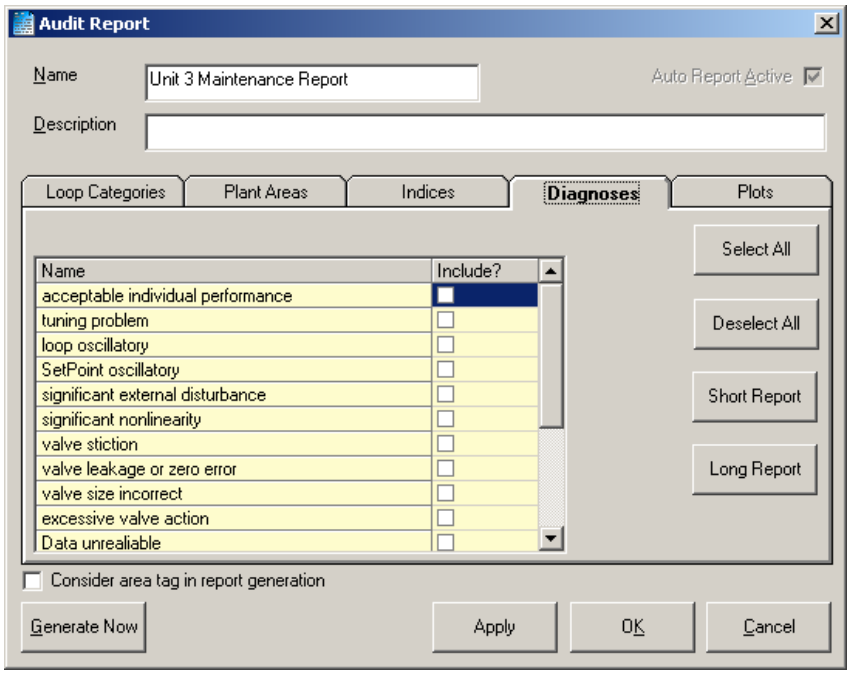


Figure 135 - Report Configuration: Diagnoses

Finally the `Plots` tab, allows the user to select critical loops (up to 10) for which a Detailed Loop report must be generated.

ABB. All rights reserved

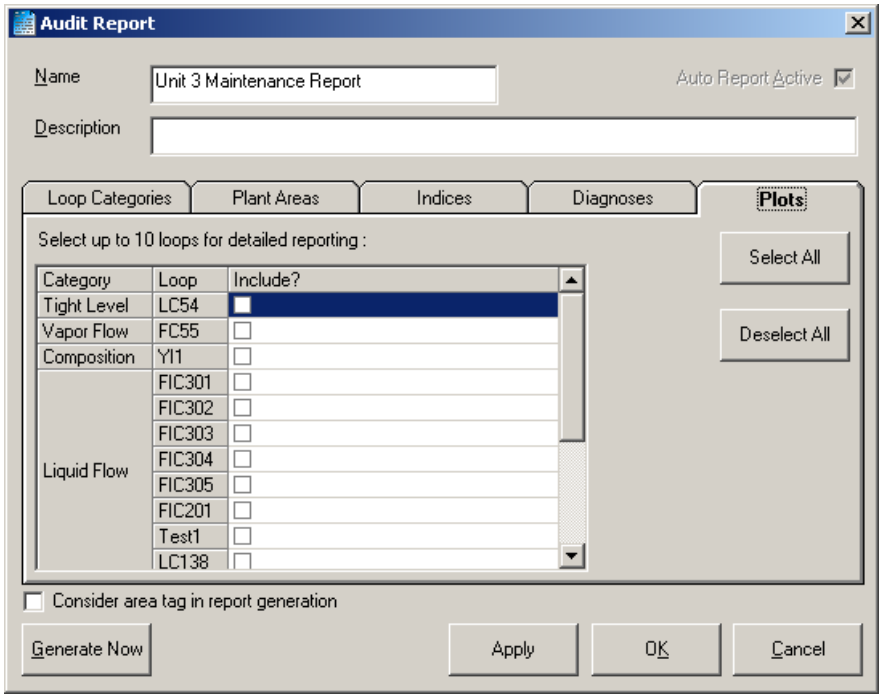


Figure 136 - Report Configuration: Detailed Loop Report

A checkbox in the lower-left corner allows the user to specify if the report should include area tags (see §6.4.2) in report generation.

6.4.2 Area Tag Reports

LPM is endowed with a sophisticated facility able to recognize and discriminate among different process operating conditions. Each Area can be configured to have an Area Classification Tag, as shown in Figure 111 - Area Configuration Window. By selecting the checkbox in the lower-left corner, the area configuration window gets the following enhanced form:

ABB. All rights reserved

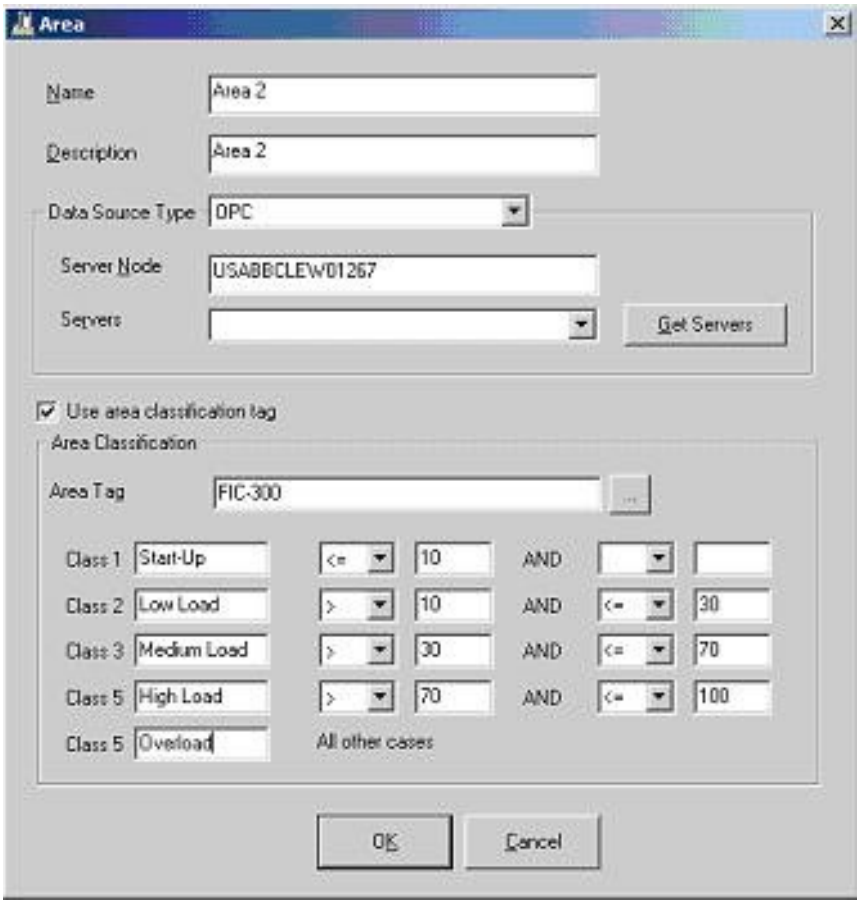


Figure 137 - Area Configuration With Area Classification Tag

The new section allows the user to define a tag (the "Area Tag"), which will act as an operation mode identifier, allowing classifying process operation in ranges, which are uniquely identified by its value.
In the example shown in Figure 137 - Area Configuration With Area Classification Tag, the plant has been recognized to work in four basic modes: start-up, low, medium and high load, with the federate variable (FIC-300) acting as the actual identifier. The configuration window allows to clearly introducing the ranges, which will identify the operational statuses.

LPM is able to accurately recognize and classify incoming auditing data in the several classes and to generate reports that highlight the loops performances according to the actual production condition. If the advanced classification option is configured and activated, then the auditing results are sorted according to the classifiers defined. This means that the report is divided up in smaller reports for each of the classes. The overall ranking will still be done using all available data batches.

6.4.3 Report Generation

Each report type that has been configured and saved may be accessed from its location in the Tree under the Report label. When opening, the user may change the report configuration and resave it.

ABB. All rights reserved

The actual reports are created automatically every week and every month, as both Excel spreadsheet files and HTML files. The file names are the custom report name plus the time-date.
The user must select in which directory he wants to have all the reports stored. This is achievable by selecting “Configure Report” under the “Tools” Menu:

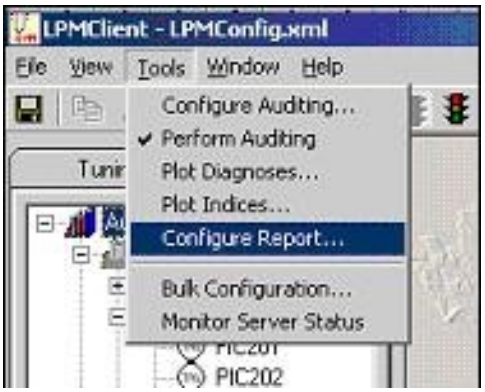


Figure 138 - Selecting Configure Report

A form like the one shown in Figure 139 - Specifying Destination Folder for Auditing Reports will appear allowing the user to properly select the favorite folder.

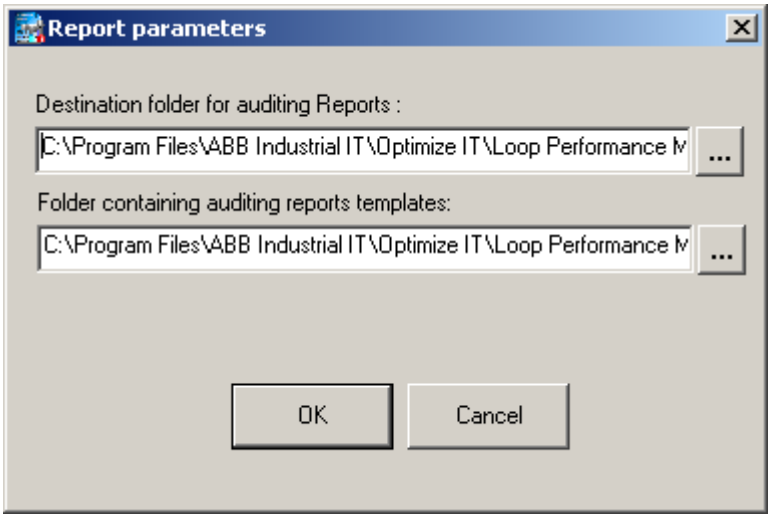
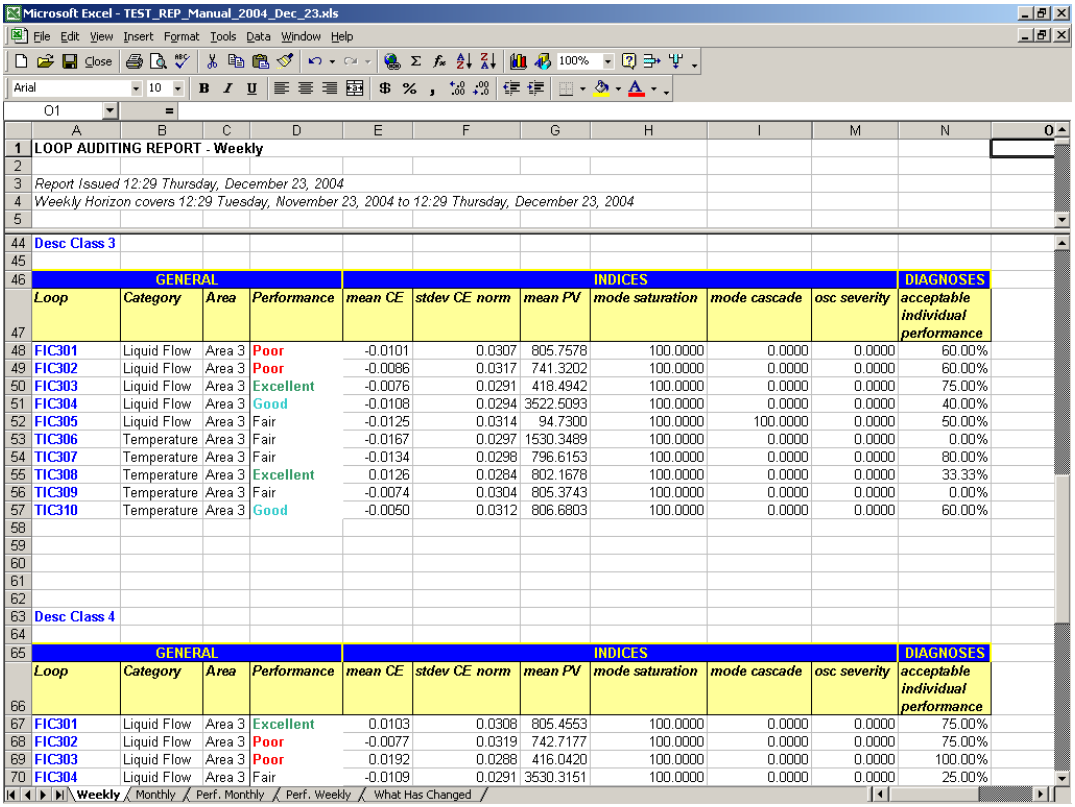


Figure 139 - Specifying Destination Folder for Auditing Reports

Figure 140 - Example of Weekly Report and Figure 141 - Example of Monthly Report shows two examples about how the reports will look like. It is possible to note that the report actually gives different evaluation for the same loops under different load conditions (Class 3, Class 4 in the figure).

ABB. All rights reserved



Microsoft Excel - TEST_REP_Manual_2004_Dec_23.xls

LOOP AUDITING REPORT - Weekly

Report Issued 12:29 Thursday, December 23, 2004

Weekly Horizon covers 12:29 Tuesday, November 23, 2004 to 12:29 Thursday, December 23, 2004

Desc Class 3

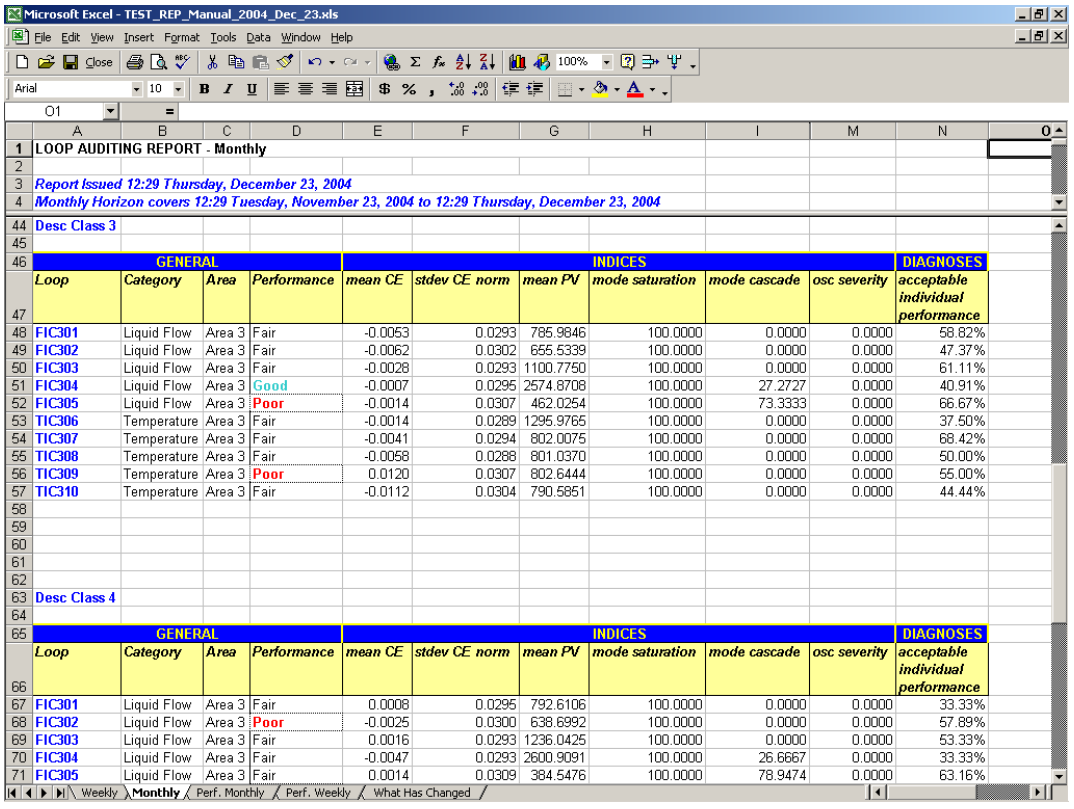
GENERAL				INDICES						DIAGNOSES
Loop	*Category*	*Area*	*Performance*	*mean CE*	*stdev CE norm*	*mean PV*	*mode saturation*	*mode cascade*	*osc severity*	*acceptable individual performance*
FIC301	Liquid Flow	Area 3	Poor	-0.0101	0.0307	805.7578	100.0000	0.0000	0.0000	60.00%
FIC302	Liquid Flow	Area 3	Poor	-0.0086	0.0317	741.3202	100.0000	0.0000	0.0000	60.00%
FIC303	Liquid Flow	Area 3	Excellent	-0.0076	0.0291	418.4942	100.0000	0.0000	0.0000	75.00%
FIC304	Liquid Flow	Area 3	Good	-0.0108	0.0294	3522.5093	100.0000	0.0000	0.0000	40.00%
FIC305	Liquid Flow	Area 3	Fair	-0.0125	0.0314	94.7300	100.0000	100.0000	0.0000	50.00%
TIC306	Temperature	Area 3	Fair	-0.0167	0.0297	1530.3489	100.0000	0.0000	0.0000	0.00%
TIC307	Temperature	Area 3	Fair	-0.0134	0.0298	796.6153	100.0000	0.0000	0.0000	80.00%
TIC308	Temperature	Area 3	Excellent	0.0126	0.0284	802.1678	100.0000	0.0000	0.0000	33.33%
TIC309	Temperature	Area 3	Fair	-0.0074	0.0304	805.3743	100.0000	0.0000	0.0000	0.00%
TIC310	Temperature	Area 3	Good	-0.0050	0.0312	806.6803	100.0000	0.0000	0.0000	60.00%

Desc Class 4

GENERAL				INDICES						DIAGNOSES
Loop	*Category*	*Area*	*Performance*	*mean CE*	*stdev CE norm*	*mean PV*	*mode saturation*	*mode cascade*	*osc severity*	*acceptable individual performance*
FIC301	Liquid Flow	Area 3	Excellent	0.0103	0.0308	805.4553	100.0000	0.0000	0.0000	75.00%
FIC302	Liquid Flow	Area 3	Poor	-0.0077	0.0319	742.7177	100.0000	0.0000	0.0000	75.00%
FIC303	Liquid Flow	Area 3	Poor	0.0192	0.0288	416.0420	100.0000	0.0000	0.0000	100.00%
FIC304	Liquid Flow	Area 3	Fair	-0.0109	0.0291	3530.3151	100.0000	0.0000	0.0000	25.00%

Weekly / Monthly / Perf. Monthly / Perf. Weekly / What Has Changed

Figure 140 - Example of Weekly Report



Microsoft Excel - TEST_REP_Manual_2004_Dec_23.xls

LOOP AUDITING REPORT - Monthly

Report Issued 12:29 Thursday, December 23, 2004

Monthly Horizon covers 12:29 Tuesday, November 23, 2004 to 12:29 Thursday, December 23, 2004

Desc Class 3

GENERAL				INDICES						DIAGNOSES
Loop	*Category*	*Area*	*Performance*	*mean CE*	*stdev CE norm*	*mean PV*	*mode saturation*	*mode cascade*	*osc severity*	*acceptable individual performance*
FIC301	Liquid Flow	Area 3	Fair	-0.0053	0.0293	785.9846	100.0000	0.0000	0.0000	58.82%
FIC302	Liquid Flow	Area 3	Fair	-0.0062	0.0302	655.5339	100.0000	0.0000	0.0000	47.37%
FIC303	Liquid Flow	Area 3	Fair	-0.0028	0.0293	1100.7750	100.0000	0.0000	0.0000	61.11%
FIC304	Liquid Flow	Area 3	Good	-0.0007	0.0295	2574.8708	100.0000	27.2727	0.0000	40.91%
FIC305	Liquid Flow	Area 3	Poor	-0.0014	0.0307	462.0254	100.0000	73.3333	0.0000	66.67%
TIC306	Temperature	Area 3	Fair	-0.0014	0.0289	1295.9765	100.0000	0.0000	0.0000	37.50%
TIC307	Temperature	Area 3	Fair	-0.0041	0.0294	802.0075	100.0000	0.0000	0.0000	68.42%
TIC308	Temperature	Area 3	Fair	-0.0058	0.0288	801.0370	100.0000	0.0000	0.0000	50.00%
TIC309	Temperature	Area 3	Poor	0.0120	0.0307	802.6444	100.0000	0.0000	0.0000	55.00%
TIC310	Temperature	Area 3	Fair	-0.0112	0.0304	790.5851	100.0000	0.0000	0.0000	44.44%

Desc Class 4

GENERAL				INDICES						DIAGNOSES
Loop	*Category*	*Area*	*Performance*	*mean CE*	*stdev CE norm*	*mean PV*	*mode saturation*	*mode cascade*	*osc severity*	*acceptable individual performance*
FIC301	Liquid Flow	Area 3	Fair	0.0008	0.0295	792.6106	100.0000	0.0000	0.0000	33.33%
FIC302	Liquid Flow	Area 3	Poor	-0.0025	0.0300	638.6992	100.0000	0.0000	0.0000	57.89%
FIC303	Liquid Flow	Area 3	Fair	0.0016	0.0293	1236.0425	100.0000	0.0000	0.0000	53.33%
FIC304	Liquid Flow	Area 3	Fair	-0.0047	0.0293	2600.9091	100.0000	26.6667	0.0000	33.33%
FIC305	Liquid Flow	Area 3	Fair	0.0014	0.0309	384.5476	100.0000	78.9474	0.0000	63.16%

Weekly / Monthly / Perf. Monthly / Perf. Weekly / What Has Changed

Figure 141 - Example of Monthly Report

ABB. All rights reserved

Figure 142 - Example of Performance Overview Report shows an example of the overall assessment report. Again the overall assessment is class-specific: each horizontal bar represents 100% of the loops of the given category: the percentage of loops performing in each of the four possible "grades" (Poor, Fair, Good, Excellent) are highlighted by means of different colors so to give immediate feeling about loops health.

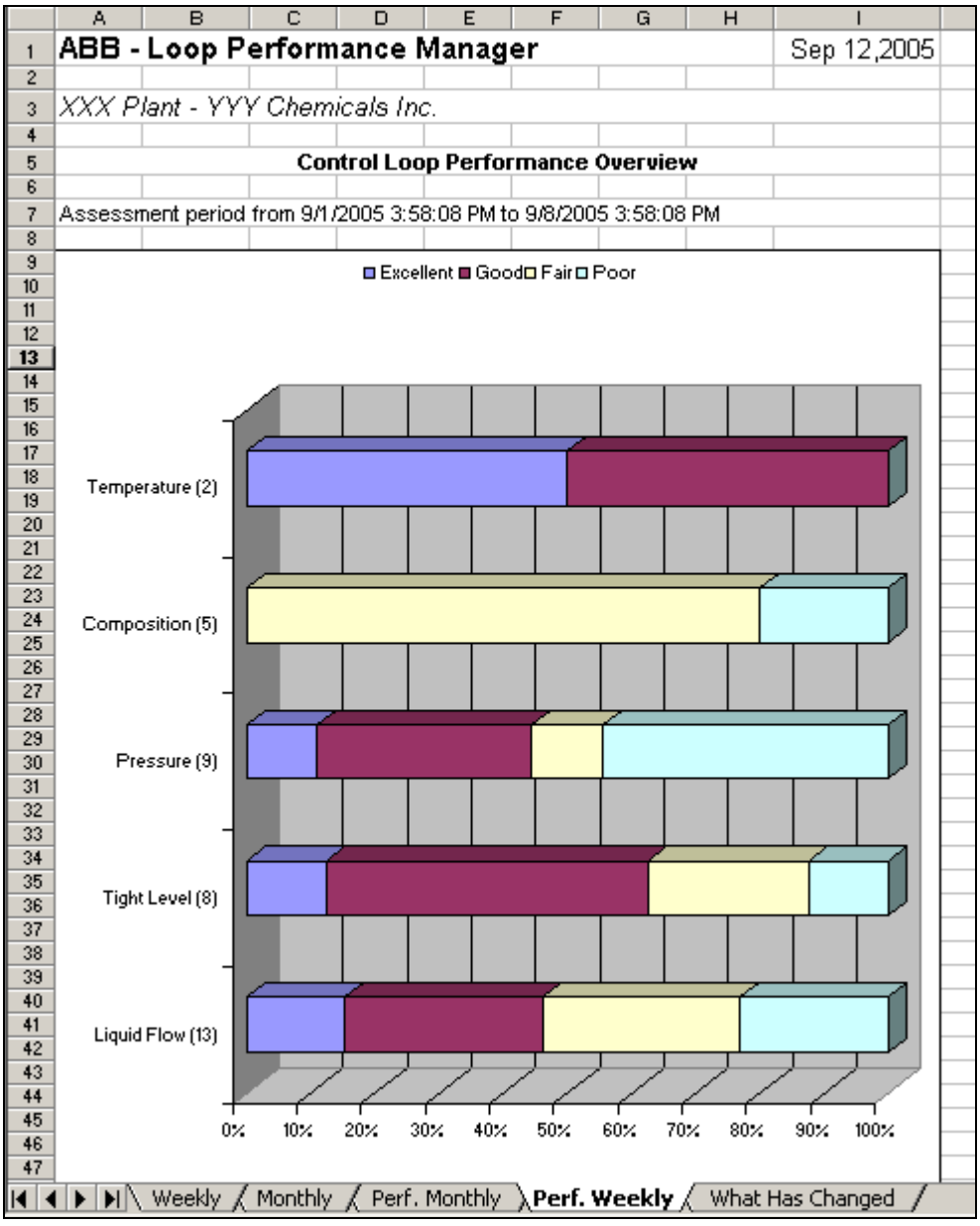
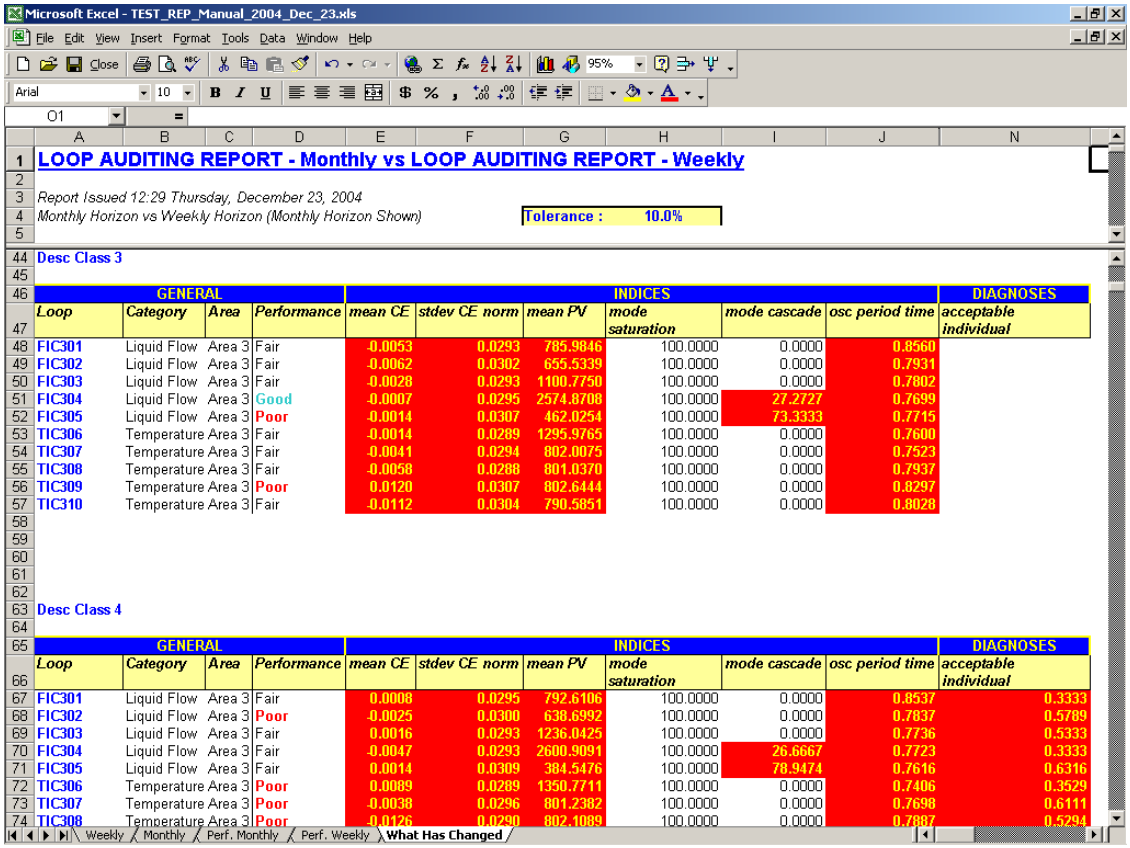


Figure 142 - Example of Performance Overview Report

In order to help the user to immediately "catch" sudden or "newborn" problems, LPM adds an additional report (significantly named "What Has Changed") where indices and diagnosis, whose value has changed more than the specified Tolerance (10% in Figure 143 – Example of the "*What Has Changed*" Report), are highlighted in red.

ABB. All rights reserved



Microsoft Excel - TEST_REP_Manual_2004_Dec_23.xls

LOOP AUDITING REPORT - Monthly vs LOOP AUDITING REPORT - Weekly

Report Issued 12:29 Thursday, December 23, 2004
Monthly Horizon vs Weekly Horizon (Monthly Horizon Shown)

Tolerance : 10.0%

Desc Class 3

	GENERAL				INDICES					DIAGNOSES
	Loop	Category	Area	Performance	mean CE	stdev CE norm	mean PV	mode saturation	mode cascade	osc period time	acceptable individual
48	FIC301	Liquid Flow	Area 3	Fair	-0.0053	0.0293	785.9846	100.0000	0.0000	0.8560	
49	FIC302	Liquid Flow	Area 3	Fair	-0.0062	0.0302	655.5339	100.0000	0.0000	0.7931	
50	FIC303	Liquid Flow	Area 3	Fair	-0.0028	0.0293	1100.7750	100.0000	0.0000	0.7802	
51	FIC304	Liquid Flow	Area 3	Good	-0.0007	0.0295	2574.8708	100.0000	27.2727	0.7699	
52	FIC305	Liquid Flow	Area 3	Poor	-0.0014	0.0307	462.0254	100.0000	73.3333	0.7715	
53	TIC306	Temperature	Area 3	Fair	-0.0014	0.0289	1295.9765	100.0000	0.0000	0.7600	
54	TIC307	Temperature	Area 3	Fair	-0.0041	0.0294	802.0075	100.0000	0.0000	0.7523	
55	TIC308	Temperature	Area 3	Fair	-0.0058	0.0288	801.0370	100.0000	0.0000	0.7937	
56	TIC309	Temperature	Area 3	Poor	0.0120	0.0307	802.6444	100.0000	0.0000	0.8297	
57	TIC310	Temperature	Area 3	Fair	-0.0112	0.0304	790.5851	100.0000	0.0000	0.8028	

Desc Class 4

	GENERAL				INDICES					DIAGNOSES
	Loop	Category	Area	Performance	mean CE	stdev CE norm	mean PV	mode saturation	mode cascade	osc period time	acceptable individual
67	FIC301	Liquid Flow	Area 3	Fair	0.0008	0.0295	792.6106	100.0000	0.0000	0.8537	0.3333
68	FIC302	Liquid Flow	Area 3	Poor	-0.0025	0.0300	638.6992	100.0000	0.0000	0.7837	0.5789
69	FIC303	Liquid Flow	Area 3	Fair	0.0016	0.0293	1236.0425	100.0000	0.0000	0.7736	0.5333
70	FIC304	Liquid Flow	Area 3	Fair	-0.0047	0.0293	2600.9091	100.0000	26.6667	0.7723	0.3333
71	FIC305	Liquid Flow	Area 3	Fair	0.0014	0.0309	384.5476	100.0000	78.9474	0.7616	0.6316
72	TIC306	Temperature	Area 3	Poor	0.0089	0.0289	1350.7711	100.0000	0.0000	0.7406	0.3529
73	TIC307	Temperature	Area 3	Poor	-0.0038	0.0296	801.2382	100.0000	0.0000	0.7698	0.6111
74	TIC308	Temperature	Area 3	Poor	-0.0126	0.0290	802.1089	100.0000	0.0000	0.7887	0.5294

Weekly | Monthly | Perf. Monthly | Perf. Weekly | What Has Changed

Figure 143 – Example of the "*What Has Changed*" Report

Finally if the user has selected one or more loops (up to 10) under the `Plots` tab (Figure 144 - Example of Detailed Loop Report), for which he/she needs more insight, additional tabs (as many as the loops specified in the Plots tab) will be added to the report, each one including 8 plots (Figure 144 - Example of Detailed Loop Report):

- Raw Data Plot (SP and PV)
- Raw Data Plot (CO)
- Histogram of control error
- Histogram of controller output
- Control Error Spectrum
- Autocorrelation and closed-loop control response
- Harris Index Sensitivity Plot
- Phase Plot

§ 6.4.4 provides a basic description of each of the above mentioned loops.

ABB. All rights reserved

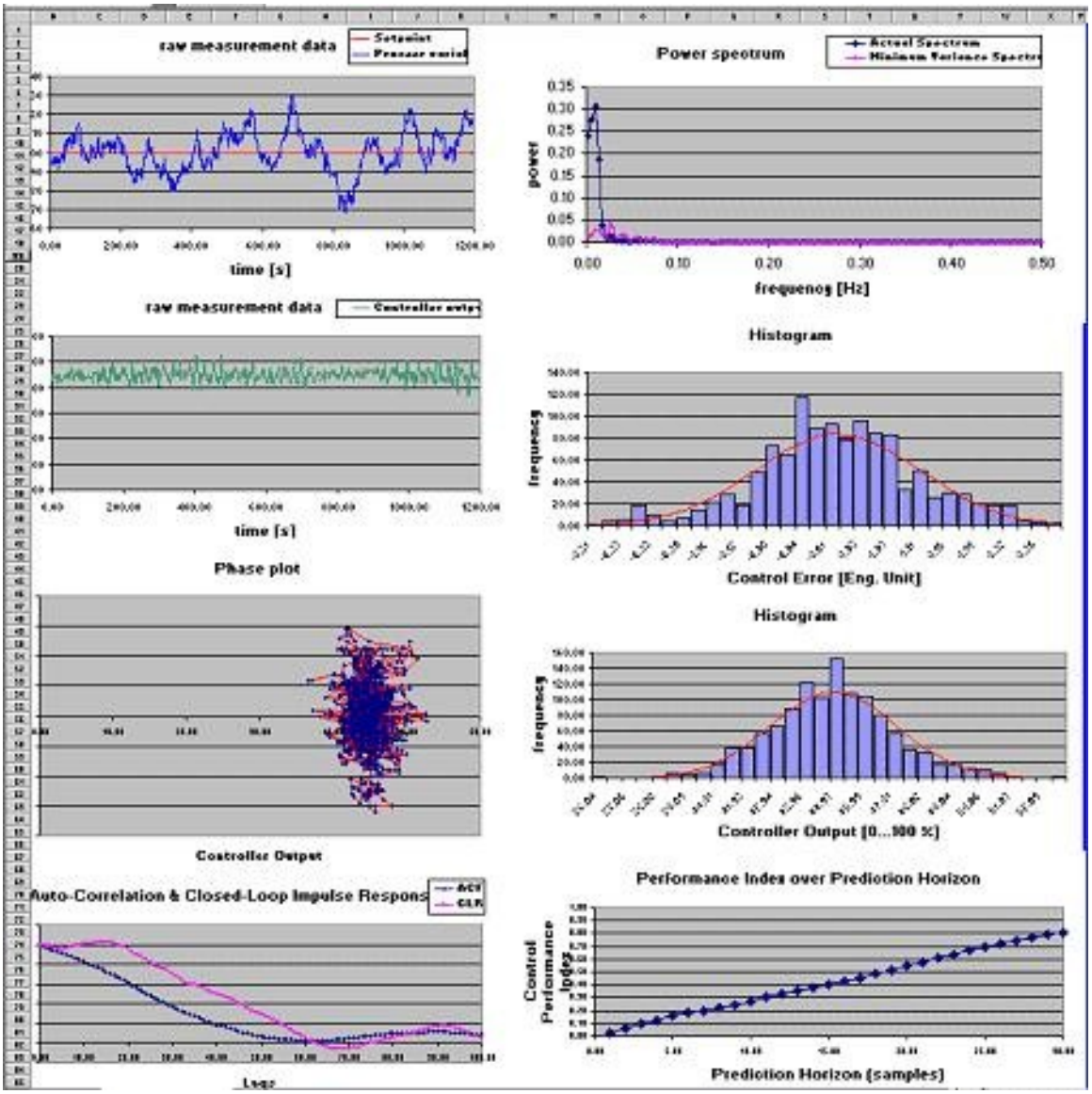


Figure 144 - Example of Detailed Loop Report

6.4.4 Loop Plots

The Detailed Loop Reports includes relevant plots that engineers typically want to see for control performance evaluations are described below. These plots are usually consulted if the experienced user wants to perform a detailed analysis after having explored both diagnoses and performance indices.

- Raw data plot - The most important plot in order to assess if the collected data set
  … is of sufficient quality (no compression, missing parts, sufficiently many samples, ...)
  … is in accordance with the analysis (e.g. oscillating, outliers, …)
- Histogram of control error and controller output - The histograms together with Gauss distributions, fitted to the actual distributions give a simple and intuitive assessment if

ABB. All rights reserved

the loop exceeds normal statistic variation significantly. Oscillations, asymmetrical distribution and off-sets are readily recognized.

- Control error Spectrum - The spectrum of the control error signal is presented. This plot represents the distribution of frequencies within the control error. The frequency of the main contributors of variance can be identified here. As a second spectrum, the minimum-variance spectrum is plotted. This spectrum represents the control error spectrum that could be achieved under minimum-variance control. Or, rather, since LPM does not benchmark control loops by their individual deadtime but by a loop category-dependent class benchmark, the second spectrum re-presents the spectrum that the benchmark controller would achieve. If both spectra are very different, the current control is far from the benchmark performance. If both spectra are close, then there is little room for improvement or the benchmark is too pessimistic. Another case is when both spectra differ in certain frequency ranges. The plots can then be used to see which frequencies contribute to the variance increase in the controller error and if the controller could remove them potentially.
- Auto-correlation and closed-loop impulse response - This plot presents the auto-correlation of the control error. This plot is usually used by control experts to quickly assess controller performance. The auto-correlation signal should decrease to zero as fast as possible. The longer this signal is different from zero and the larger the deviation is, the worse the control performance usually is. A plot related to auto-correlation is the closed-loop impulse response, estimated when computing the Harris index. The algorithms compute the time-series model for the controller error. The closed-loop impulse response from the unknown disturbances to the process variable is obtained from a series expansion of the estimated time-series model. Both auto-correlation function and closed-loop impulse response have similar shapes. The closed-loop impulse response is directly related to the Harris performance index. It is a heuristic illustration of the closeness of the current control to the benchmark performance.
- Harris index sensitivity plot - The choice of the prediction horizon for the Harris index computation is made for a complete class of loop categories (flow, pressure, etc). Of course, this benchmark is not the optimal one for any kind of loop. Therefore, it is important to know how much the index can be trusted. The sensitivity plot presents the value of the Harris index for a whole range of possible prediction horizons. If the index curve is high (low) around the current prediction horizon, the performance is good (bad), no matter if the horizon is slightly changed or not. In this case, the index is not sensitive towards the horizon. This is the most common case and a motivation to use this concept. If the index curve is very steep around the current prediction horizon, then the performance index result needs to be used with care since a slight change of the benchmark can change the assessment drastically.
- Phase plot - The phase plot shows control error and controller output plotted against each other. Such plot is useful to present periodic and nonlinear disturbances in the data that may not be easily visible in the raw data plots. Oscillations are usually visible as elliptic plots and nonlinearities may show abrupt edges in the phase plot. Good control usually is presented in a cloud of samples that is centered on the operating point.

ABB. All rights reserved

6.5 *Interactive Reports*

This feature allows the user to generate on-demand reports in the LPM Client. The reports are generated based on user selections and are presented within the client workspace.
Interactive reports can be accessed by clicking on *Tools/Interactive reports* entry in the menu. As the menu entry is selected, a selection form is presented.

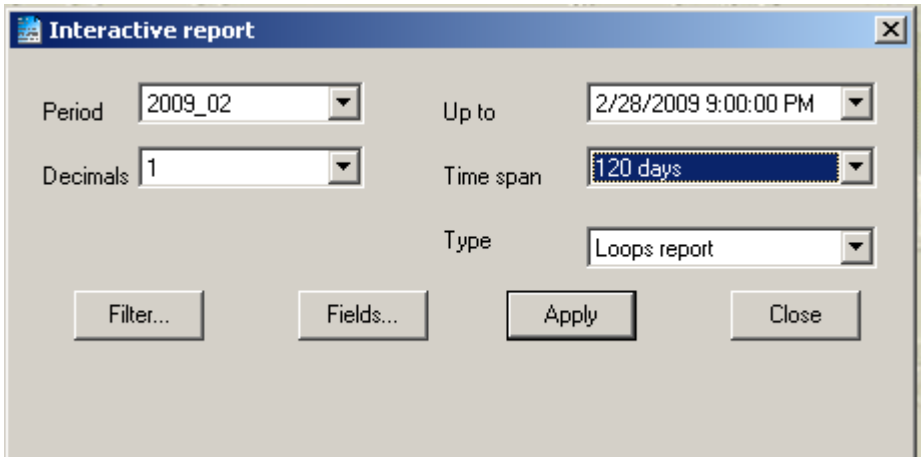


Figure 145 – Interactive Report selection form

This form allows the user to enter the report properties. The following items can be selected:

Item	Description	Valid values	Notes
Period	Month	2009_01 2009_02 Etc.	In case of multi month reports, the last month covered
Up to	Last date covered	The combo box presents valid values extracted for the selected month	The last batch data covered in the report
Time span	Report duration	Valid values go from 1 day to 365 days. Snapshot report is also available	
Type	Report type	Loops report Aggregate report	
Decimals	Number of decimals	1-10	

Table 7 – Interactive Report properties

6.5.1 *Loops report*

6.5.2 *Aggregate report*

The aggregate reports present information on aggregate Area and Category performances with regards to Diagnosis, Preconditions and Ranks. The following information is presented.

ABB. All rights reserved

6.5.2.1 Area Overview Tab

The form presents the aggregate averages based on the Area the loops belong to.

Item	Description	Type
Area averages for Diagnosis, Grids and Ranks	Computed averages on loops belonging to the Area	Table
Area ranks distributions	Rank distribution (excellent, good, etc.) per area	Pie chart
KPI averages	Computed averages for a selected KPI	Histogram chart

Table 8 - Area aggregate report items

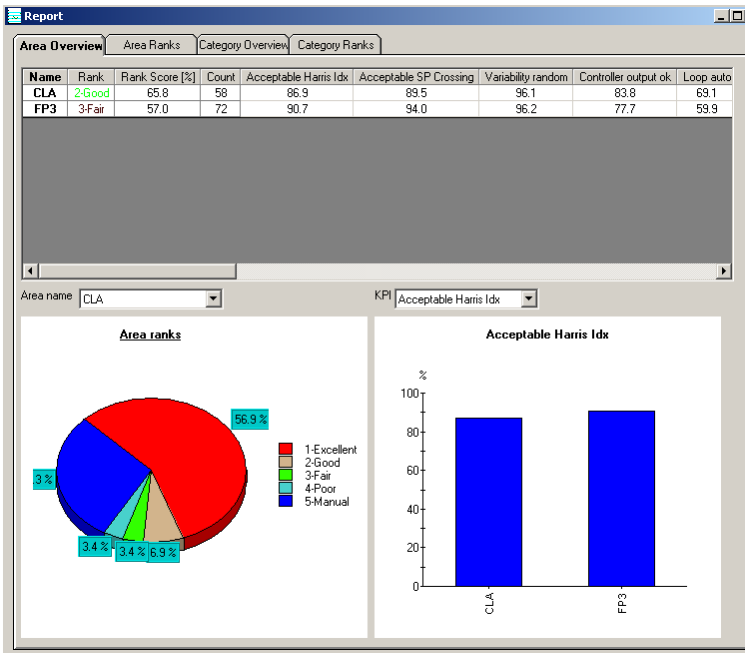


Figure 146 – Aggregate report, area overview

6.6 *Advanced Auditing Settings*

This feature allows the user to fine tune the auditing indices, diagnosis and Rank calculations. The relevant configuration form can be prompted by selecting *Tools/Advanced Auditing Settings* menu entry. Several tuning factors are available and are organized into multiple tabs of the configuration form. Details on each tab are given in the next paragraphs.

Notice: the settings presented in this form are key in the computation of auditing indices, diagnosis and ranks. Modifications applied to these settings could impact heavily auditing calculations. Changes in these settings should be implemented only by experienced users.

ABB. All rights reserved

6.6.1 Prediction Horizon Tab

This tab presents information on each category setting for Prediction Horizon (PH). The prediction horizon is used as estimated dead time in the Harris Index calculation and is therefore very important to compute the loop Performance Index. Two values are presented, the PH for normal and the PH master loops in a cascade configuration. The engineering unit is seconds.

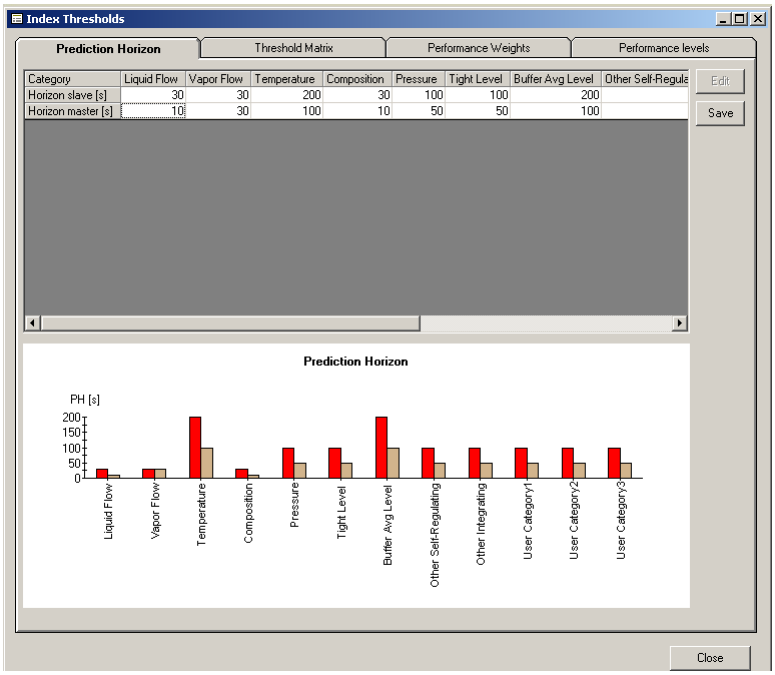


Figure 147 – Prediction Horizon settings

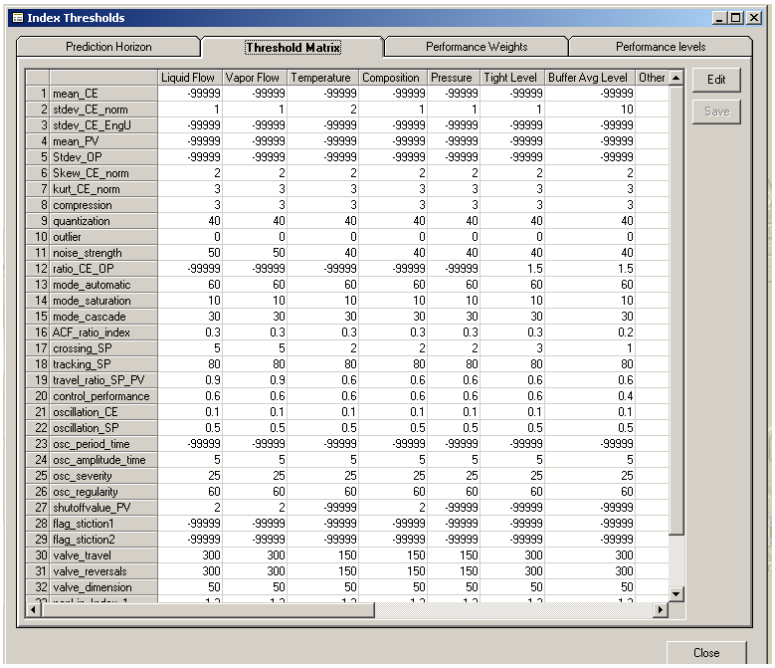
Changes to the PH values can be implemented by pressing the Edit pushbutton on the right side of the form.

6.6.2 Threshold Matrix Tab

The threshold matrix tab allows the user to inspect and modify threshold settings used to compute diagnosis/preconditions from indices.
These values are used to fine tune the threshold limit related to a single index in the diagnosis computation. Indices values below the threshold(s) do not generate the relevant diagnosis, while values higher than the threshold do set the diagnosis to true. These thresholds are quite important in determining when an alarm/indication of a potential fault is given.

Some indexes are no used for diagnosis/preconditions calculation. A simple example is the process value average value. For these indices a value of -99999 is presented as threshold.

ABB. All rights reserved



		Liquid Flow	Vapor Flow	Temperature	Composition	Pressure	Tight Level	Buffer Avg Level	Other
1	mean_CE	-99999	-99999	-99999	-99999	-99999	-99999	-99999	
2	stdev_CE_norm	1	1	2	1	1	1	10	
3	stdev_CE_EngU	-99999	-99999	-99999	-99999	-99999	-99999	-99999	
4	mean_PV	-99999	-99999	-99999	-99999	-99999	-99999	-99999	
5	Stdev_OP	-99999	-99999	-99999	-99999	-99999	-99999	-99999	
6	Skew_CE_norm	2	2	2	2	2	2	2	
7	kurt_CE_norm	3	3	3	3	3	3	3	
8	compression	3	3	3	3	3	3	3	
9	quantization	40	40	40	40	40	40	40	
10	outlier	0	0	0	0	0	0	0	
11	noise_strength	50	50	40	40	40	40	40	
12	ratio_CE_OP	-99999	-99999	-99999	-99999	-99999	1.5	1.5	
13	mode_automatic	60	60	60	60	60	60	60	
14	mode_saturation	10	10	10	10	10	10	10	
15	mode_cascade	30	30	30	30	30	30	30	
16	ACF_ratio_index	0.3	0.3	0.3	0.3	0.3	0.3	0.2	
17	crossing_SP	5	5	2	2	2	3	1	
18	tracking_SP	80	80	80	80	80	80	80	
19	travel_ratio_SP_PV	0.9	0.9	0.6	0.6	0.6	0.6	0.6	
20	control_performance	0.6	0.6	0.6	0.6	0.6	0.6	0.4	
21	oscillation_CE	0.1	0.1	0.1	0.1	0.1	0.1	0.1	
22	oscillation_SP	0.5	0.5	0.5	0.5	0.5	0.5	0.5	
23	osc_period_time	-99999	-99999	-99999	-99999	-99999	-99999	-99999	
24	osc_amplitude_time	5	5	5	5	5	5	5	
25	osc_severity	25	25	25	25	25	25	25	
26	osc_regularity	60	60	60	60	60	60	60	
27	shutoffvalue_PV	2	2	-99999	2	-99999	-99999	-99999	
28	flag_stiction1	-99999	-99999	-99999	-99999	-99999	-99999	-99999	
29	flag_stiction2	-99999	-99999	-99999	-99999	-99999	-99999	-99999	
30	valve_travel	300	300	150	150	150	300	300	
31	valve_reversals	300	300	150	150	150	300	300	
32	valve_dimension	50	50	50	50	50	50	50	

Figure 148 – Threshold matrix settings

6.6.3 Rank and Rank score concepts

User is allowed define the weights used for each Precondition in the Rank Score calculation. Each category has a different sets of weights used in the Rank Score calculation.

The Precondition list is given in the following table

Item #	Description	Notes
1	Acceptable Harris Index	
2	Acceptable Setpoint Crossing	
3	Variability Random	
4	Controller output in range	
5	Loop in auto mode	
6	Acceptable cascade performance	Applies to cascades only
7	Acceptable speed	

Table 9 – Preconditions list

The Rank Score is computed following this procedure:

- The precondition average is computed over the report time period for each precondition. This value is expressed as %. Inconclusive values[4] are not considered in the average. If no value is available, the average is set to inconclusive.

[4] A precondition can either be True (1), False (0) or Inconclusive. (-1)

ABB. All rights reserved

- All precondition averages are then weighted using the precondition weights presented in this form. Each category has different settings for the weights. Weights are expressed in %.

The result, known as Rank Score is then compared with ranges to determine the resulting Rank. The default rank ranges are presented in the next table.

Rank	Rank Score Range	Notes
Excellent	>= 90 %	
Good	>=80 % <90 %	
Fair	>= 50 % <80 %	
Poor	<50 %	
Manual	Loop is in manual	Rank available only in Interactive Reports

Table 10 – Rank list

6.6.4 Performance Weights Tab

This tab allows the user to inspect and fine tune each category performance weights. In this way a user could e.g. increase the importance of a certain precondition against another precondition. The sum of all weights should equal 100%. If not, LPM will rescale to 100% prior to weights save.

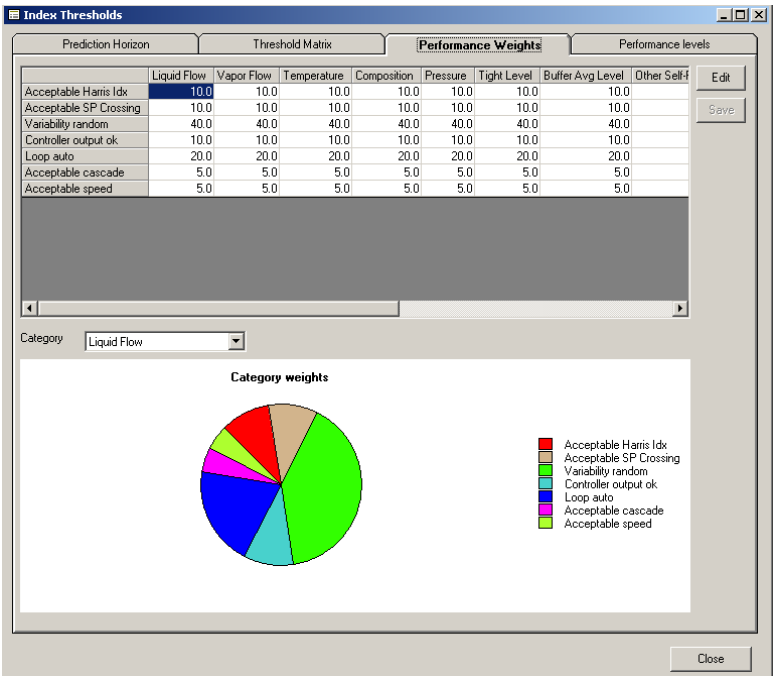


Figure 149 – Performance Weights Tab

6.6.5 Performance Levels Tab

This tab allows the user to inspect and fine tune the levels used to compute loop ranks. Default values for the levels are presented in the previous table.

ABB. All rights reserved

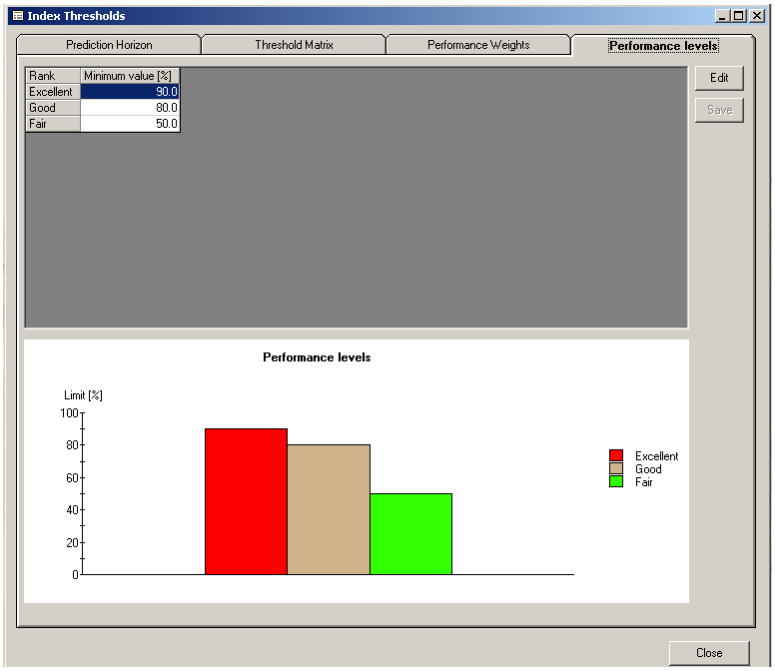


Figure 150 – Rank levels tab

6.7 *Monitor Server Status*

LPM is endowed with a facility designed to allow the user to always be informed about the status of the operations performed by the routines running automatically and managed by the LPM Server module. In order to launch the Monitor Server Status, the user must select "`Monitor Server Status`" in the "`Tools`" menu as described in Figure 151 – Monitor Server Status:

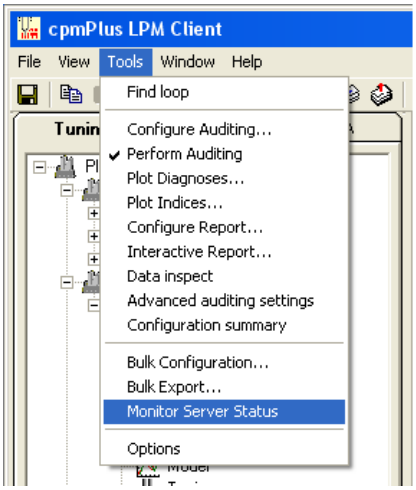


Figure 151 – Monitor Server Status

This will pop-up a form like the one shown below, structured in three tabs:

- Schedule
- Statistics

ABB. All rights reserved

- Tag Values

In addition to the above, it contains “Auditing Collections List”

The “Auditing Collections List” shows the list of auditing collections (both batch and continuous) in a list box. It contains information with columns like: File Name, Collection type, Collection Start and End date, etc.

The Schedule tab consists of a line graph showing the collection progress in time. It can be noted that when you select a collection in the Auditing collections list, the corresponding collection in the line-graph is highlighted by getting thicker. It is as shown in Figure 152 – Status Monitor: LPM Auditing collections list and Schedule below.

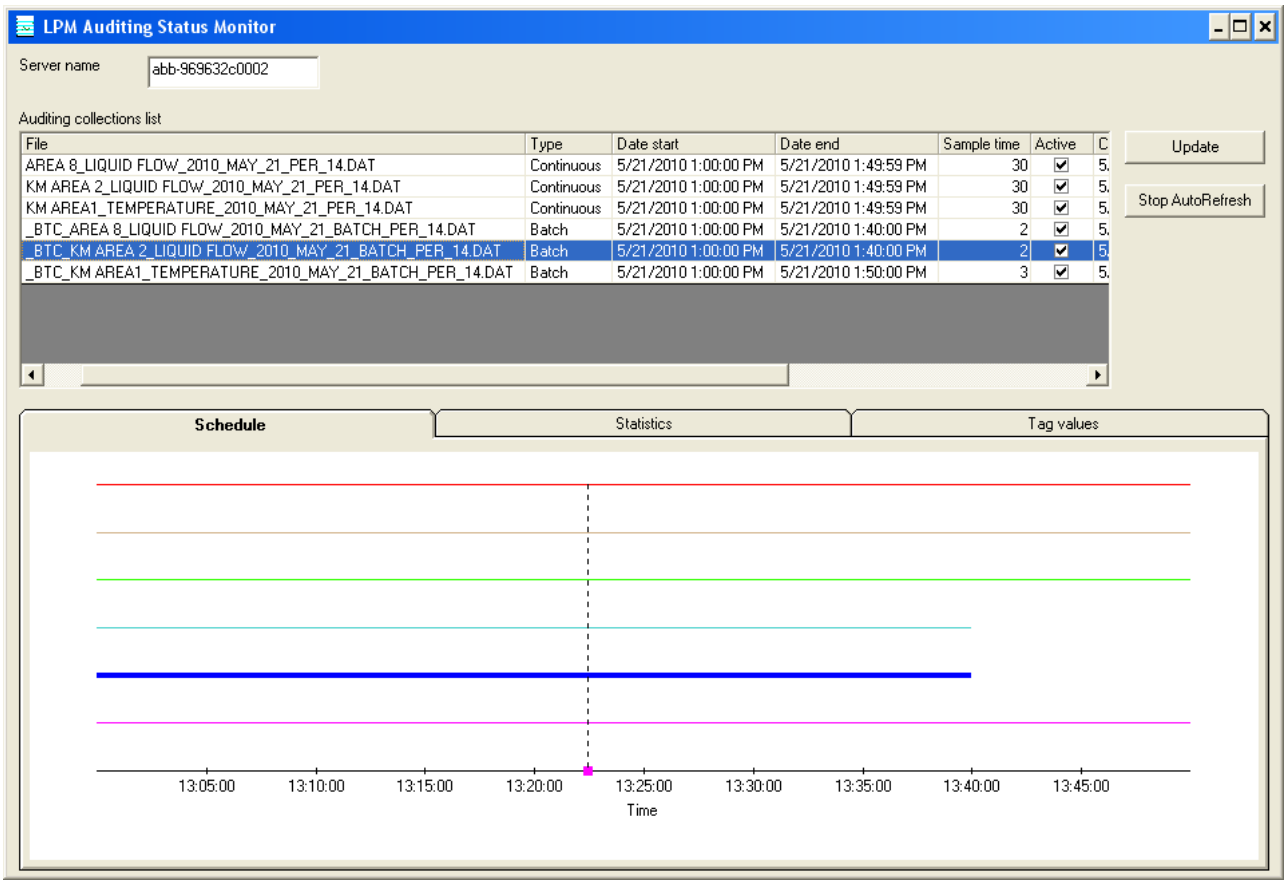


Figure 152 – Status Monitor: LPM Auditing collections list and Schedule

The Statistics tab provides graphical progressive information on the remaining time and the File size of the different collections. See Figure 153 - Status Monitor: Statistics below

ABB. All rights reserved

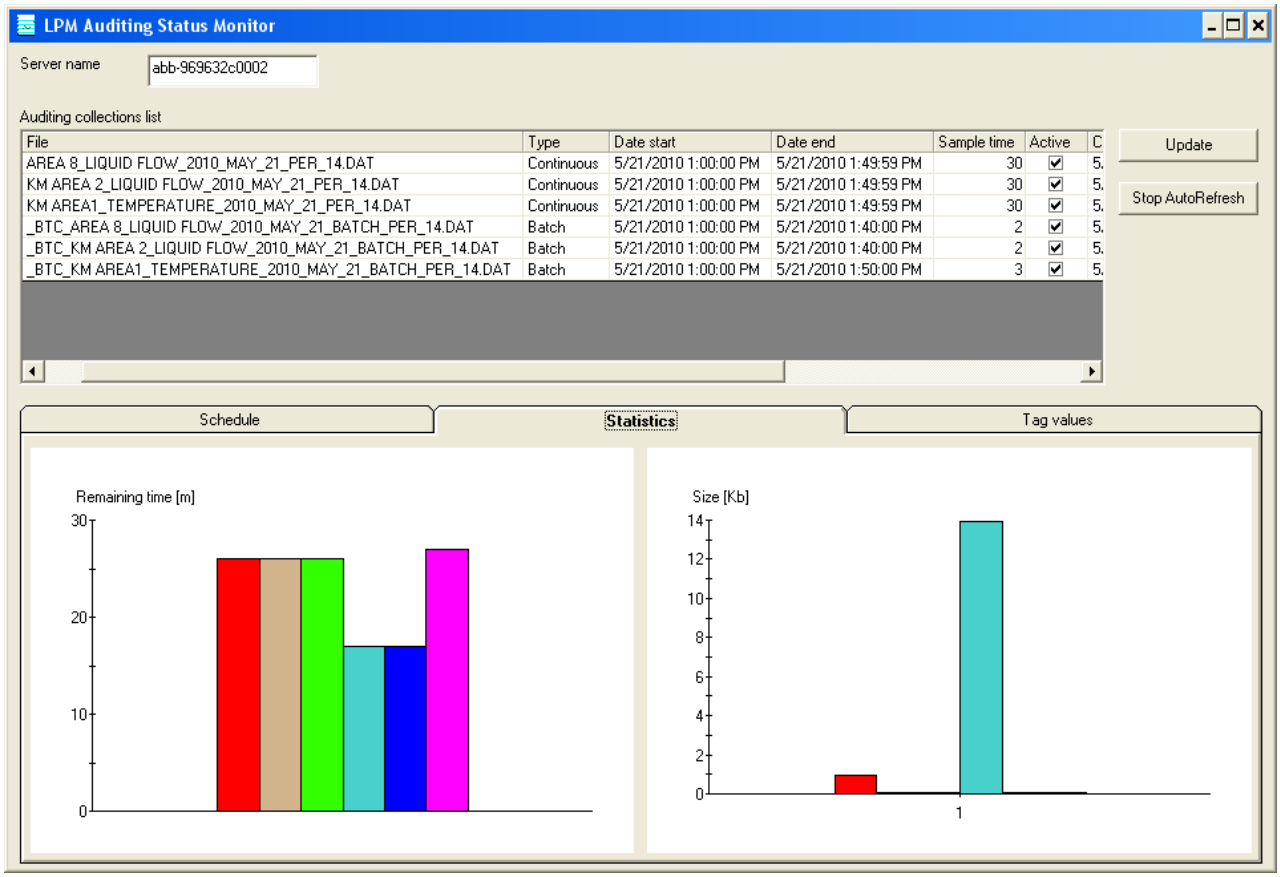


Figure 153 - Status Monitor: Statistics

Finally the "Tag values" Tab contains information about the Auditing collection Server and a command button: Update Values. When a collection is selected from the "Auditing collections list", and this "Update values" [5] button is pressed, the grid: "Last sampled values" will be populated with the current values. This is just to verify the current read sample by Auditing server. See Figure 154 - Status Monitor: Tag values below.

[5] The Update Values may not work perfectly for a Collection that has Auditing data source as KM. This is due to the fact that during Auditing collection in progress for a KM related collection, the raw data is read every five minutes irrespective of actual sampling rate. (Refer to section: 7.5.2) You will be able to read the values only in case the Auditing Collection has just read the KM values, and you have clicked on the "Update Values" button at the same instant. This may be a challenging task, hence for viewing the last value read, please use the "View Auditing Data" feature.(as shown in Figure: 175).

ABB. All rights reserved

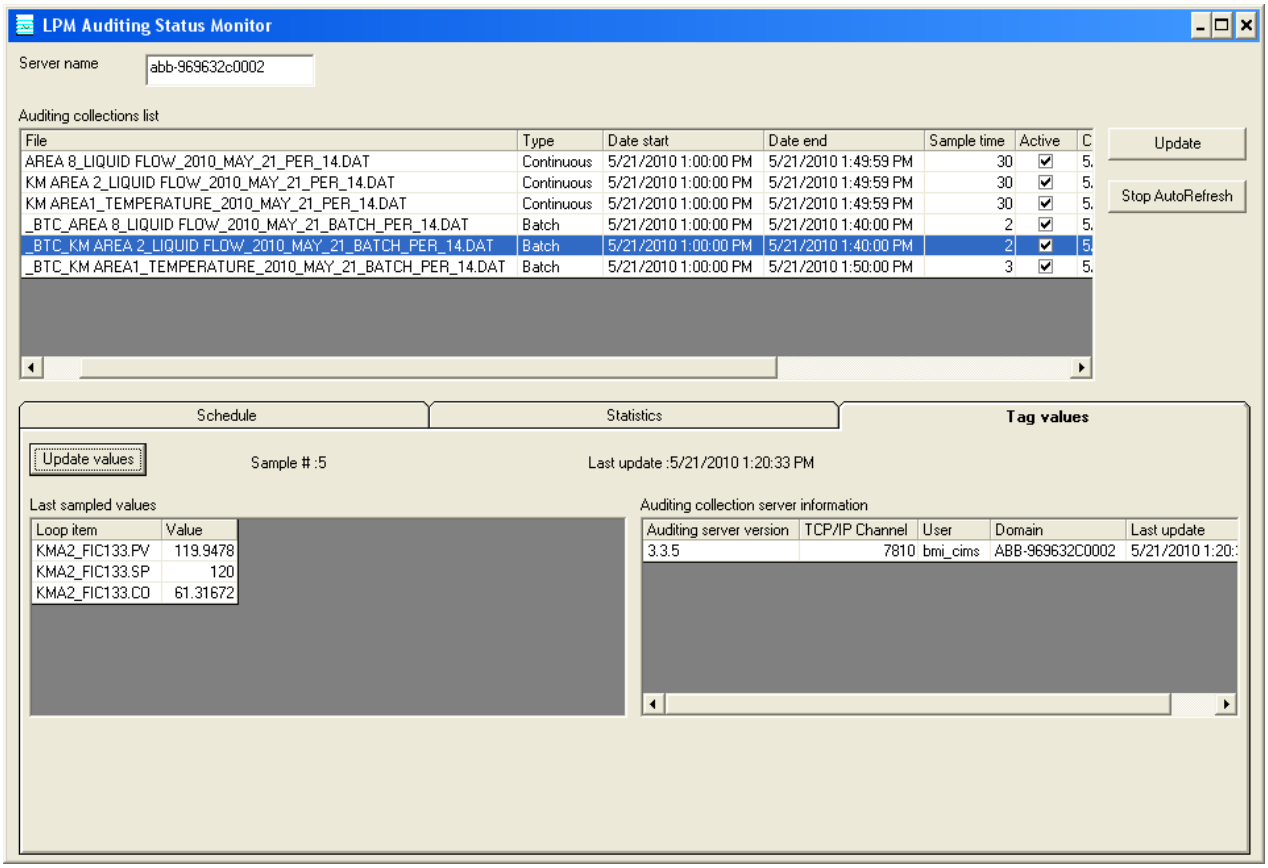


Figure 154 - Status Monitor: Tag values

The auditing Status Monitor also contains two more buttons namely: “Update” and “Stop Auto Refresh”. The Status monitor will update all the information displayed every 2 minutes automatically when the Status Monitor Screen is active and displayed on screen. If you do not want to update automatically then you can press the button- “Stop Auto Refresh”. The button then changes to “Start Auto Refresh”. Later if you want again the Status Monitor to do auto refresh, you can click once again the button- “Start Auto Refresh”.

While you have stopped the auto refresh in the monitor to observe the data, and then you want to just to get updated data at once. This can be achieved by pressing “Update” button.
Thus with this “Update” button you can update the status monitor manually at any point of time and get the latest status on Auditing progress.

Overall the LPM Auditing Status Monitor facility grants a valuable tool for auditing troubleshooting and self-diagnostics.

6.8 *LPM Web Client*

Good engineering practice requires that LPM is hosted on a PC which is placed in or close to the control room. This obviously could be a nuisance for the control or the maintenance engineer who might be interested in accessing LPM generated information often and from PCs located more

ABB. All rights reserved

comfortably for him (e.g. in his/her own office). LPM Web Client is a utility created just to make LPM even more useful and accessible from as many computers as needed.

Exploiting the IIS Microsoft technology, LPM Web Server is able to remotely support the most common monitoring and even execution commands. To be launched, the hosting client PC needs to have Microsoft Internet Explorer™ installed. Inserting in the "`Address`" area the proper location (as shown, for example in Figure 155 – Web Client Address), the user will access an entry page like the one shown in Figure 156 – LPM Web Client Login:

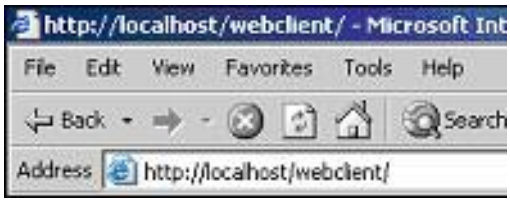


Figure 155 – Web Client Address

In the case shown in Figure 155 – Web Client Address the connection is addressed to a server web located on the same PC (this is the reason for "localhost"). When connecting on a network PC the PC name must be obviously inserted.

Figure 156 – LPM Web Client Login

Providing the Username and Password, which should have been configured as explained in§ 6.8.1, the user will access the LPM Web Server homepage as shown in Figure 157 – LPM Web Homepage:

At the first installation the default Username and password are:

- usr, usr for Normal User right
- amd, adm for Power User right
- admin, admin for Administrator right

ABB. All rights reserved

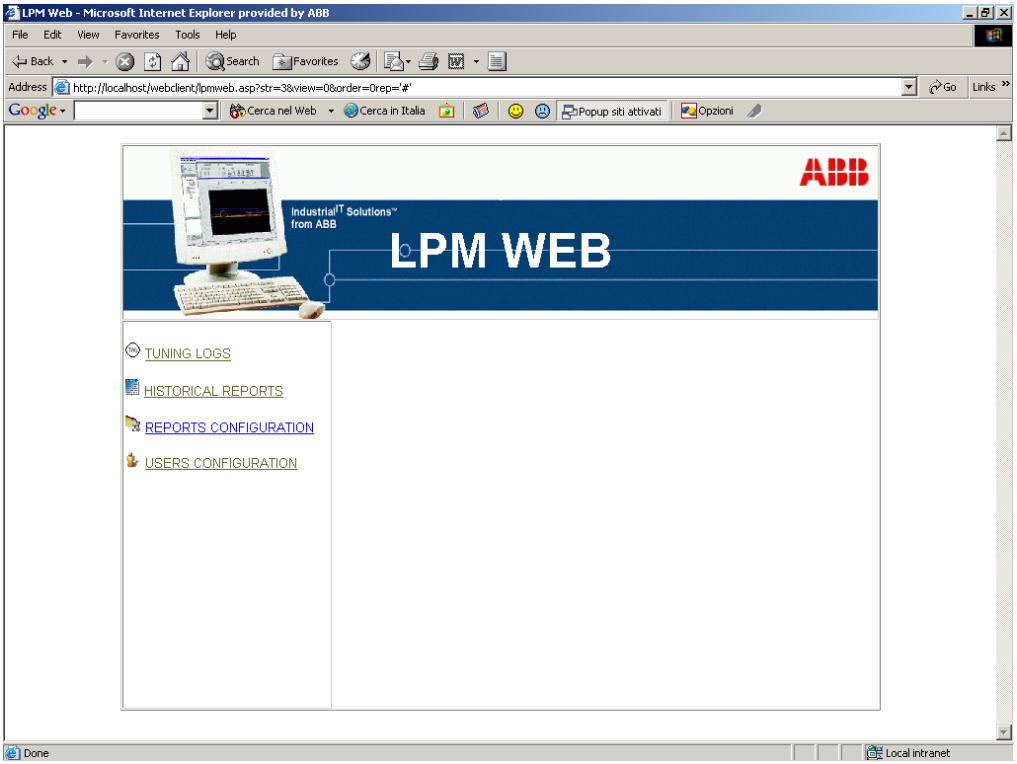


Figure 157 – LPM Web Homepage

The first three options on the left pane allow, respectively, to remotely access:

- **TUNING LOGS**: tuning log files
- **HISTORICAL REPORTS**: historical report actually stored on disk
- **REPORTS CONFIGURATION**: modify report configuration and generate reports on-demand.

Selecting the Loop option (Figure 158 - Remotely Accessing Log Files), the user can open and investigate all the tuning logs that have been saved for all the configured loops, by simply clicking on the correspondent hyperlink.

ABB. All rights reserved

Figure 158 - Remotely Accessing Log Files

Figure 159 - Remotely Accessing Historical Reports

Selecting the Reports option (Figure 159 - Remotely Accessing Historical Reports), the user can open and investigate all the auditing reports (both in HTML or Excel-based) that have been saved for all the configured loops, by simply clicking on the correspondent hyperlink.

Selecting the Configured Reports option (Figure 160 - Remote Report Configuration), the user can delete, open or modify all the auditing reports logs that have been saved for all the configured loops, by simply clicking on the correspondent icon (respectively).

ABB. All rights reserved

Figure 160 - Remote Report Configuration

When the user desires to modify an existing Report template, a form like the one shown in Figure 161 – Creating/Modifying Report Configuration is made available so that all the options available on the actual LPM Client are duplicated even on the remote PC.

Finally the user can remotely create a brand-new report just clicking on the

button: again a form like the one in Figure 161 – Creating/Modifying Report Configuration will appear to allow to insert the desired options.

ABB. All rights reserved

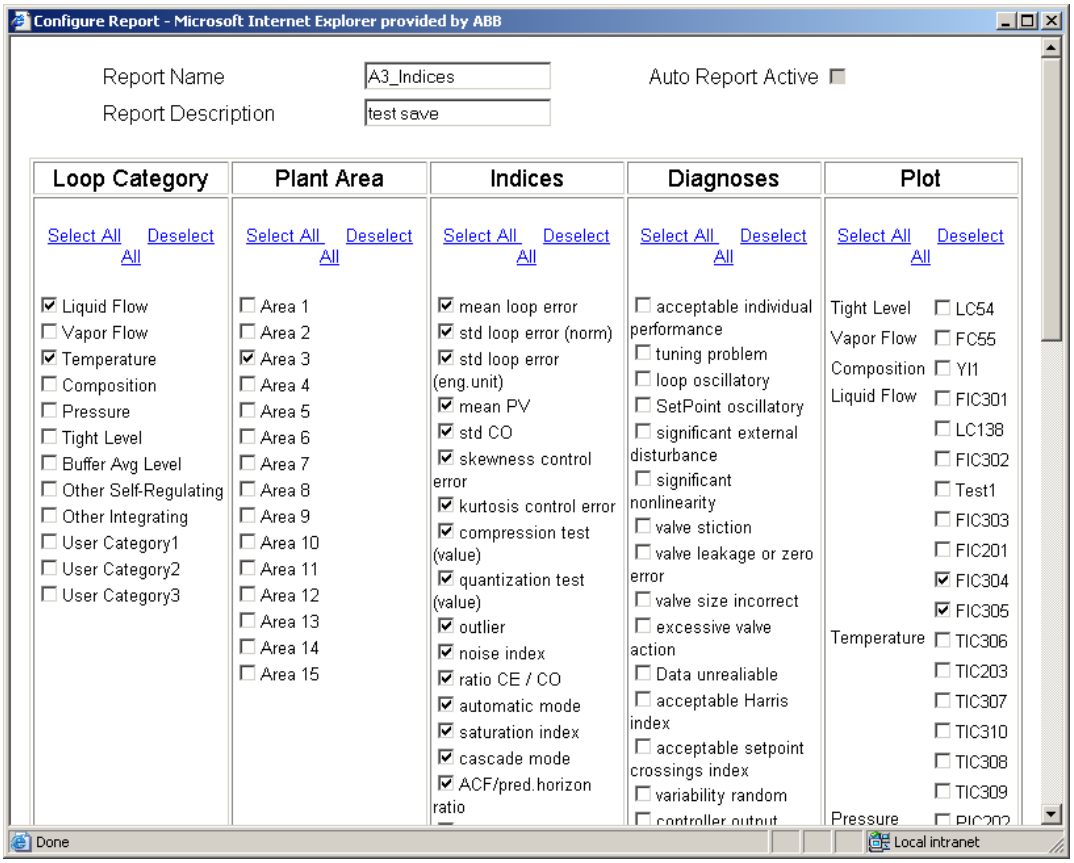


Figure 161 – Creating/Modifying Report Configuration

If the user has been given the Administrative rights a fourth option ("`Configure Users`") will appear on the screen as shown in Figure 157 – LPM Web Homepage. Selecting this option, a new form like the one shown in Figure 162 - LPM Web with Administrative Rights will appear.
From here the user can:

- Add a new user (the "`Add User`" hyperlink)
- Modify configured users' profile (the "`Configured Users`" hyperlink)
- Return to the LPM Web Server homepage (the "`Return to LPM Page`" hyperlink)

When adding a new user (Figure 163 – LPM Web: Adding Users) there are three possible profiles:

- Normal User, able to access Tuning Logs and Historical Reports.
- Power User, able to access Tuning Logs, read Historical Reports and configure/create new reports.
- Administrator, able to access Tuning Logs, read Historical Reports, configure/create new reports and configure web user.

In each case the user must be given a unique name and Username and a correspondent Password.

ABB. All rights reserved

Figure 162 - LPM Web with Administrative Rights

Figure 163 – LPM Web: Adding Users

ABB. All rights reserved

When modifying a user profile, the administrator will be presented with a grid like the one shown in Figure 164 – LPM Web: Configured Users, where he can access the user profile by just clicking on the correspondent hyperlinks.

Figure 164 – LPM Web: Configured Users

By clicking on "`Modify`" a new Window will be opened with all user data. In this new window it is possible to modify user data and store them by clicking on "Modify User".
Use "`Remove`" link to delete a user. An example of the User modify data window is given in Figure 165 – LPM Web: Modify User.

ABB. All rights reserved

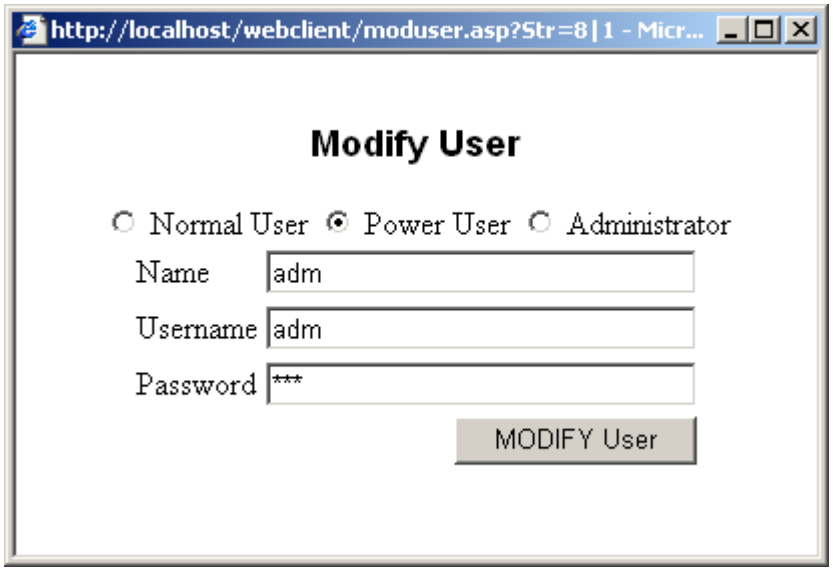


Figure 165 – LPM Web: Modify User

6.8.1 LPM Web Client Installation

Once LPM has been installed. The LPM Web Server is positioned in the folder "C:\Program Files\ABB Industrial IT\Optimize IT\Loop Performance Manager\WebClient\".
In order to have it working the Microsoft IIS (Internet Information Services) server web must be properly installed (the Microsoft Windows installation CD may be requested) and a site must be configured in the WEBCLIENT folder.
In order to Install IIS and to configure the site the following procedure must be followed:

a. **Start**
b. **Settings**
c. **Control Panel**
d. **Add/Remove Programs**
e. **Add/Remove Window Components (**Figure 166 – Add/Remove Windows Components**)**

ABB. All rights reserved

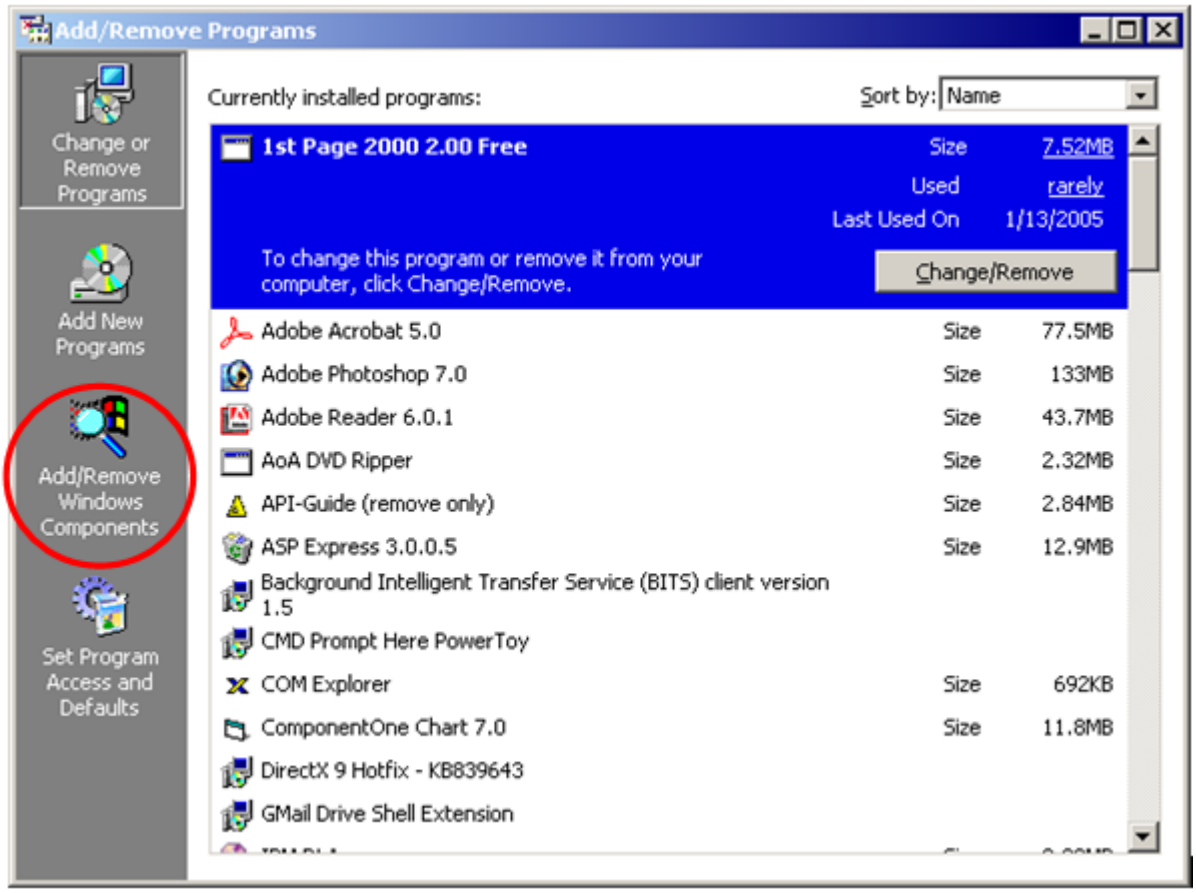

Add/Remove Programs
Change or Remove Programs
Add New Programs
Add/Remove Windows Components
Set Program Access and Defaults
Currently installed programs:
Sort by: Name
1st Page 2000 2.00 Free
Size 7.52MB
Used rarely
Last Used On 1/13/2005
To change this program or remove it from your computer, click Change/Remove.
Change/Remove
Adobe Acrobat 5.0 Size 77.5MB
Adobe Photoshop 7.0 Size 133MB
Adobe Reader 6.0.1 Size 43.7MB
AoA DVD Ripper Size 2.32MB
API-Guide (remove only) Size 2.84MB
ASP Express 3.0.0.5 Size 12.9MB
Background Intelligent Transfer Service (BITS) client version 1.5
CMD Prompt Here PowerToy
COM Explorer Size 692KB
ComponentOne Chart 7.0 Size 11.8MB
DirectX 9 Hotfix - KB839643
GMail Drive Shell Extension

Figure 166 – Add/Remove Windows Components

f. Select "**Internet Information Services (IIS)**"

ABB. All rights reserved

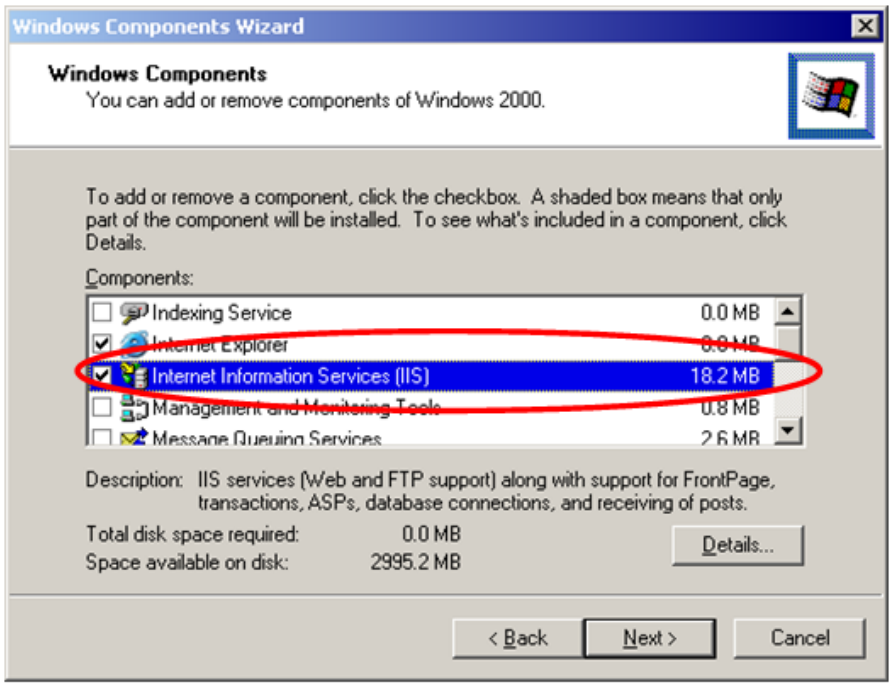

Windows Components Wizard
Windows Components
You can add or remove components of Windows 2000.
To add or remove a component, click the checkbox. A shaded box means that only part of the component will be installed. To see what's included in a component, click Details.
Components:
Indexing Service 0.0 MB
Internet Explorer
Internet Information Services (IIS) 18.2 MB
Management and Monitoring Tools 0.8 MB
Message Queuing Services 2.6 MB
Description: IIS services (Web and FTP support) along with support for FrontPage, transactions, ASPs, database connections, and receiving of posts.
Total disk space required: 0.0 MB
Space available on disk: 2995.2 MB
Details...
< Back
Next >
Cancel

Figure 167 –IIS Selection

g. Clicking on "**Next**" the **installation** procedure begins

Configuration of the shared folder:

- Right-click on the WebClient folder (C:\Program Files\ABB Industrial IT\Optimize IT\Loop Performance Manager\WebClient\)
- Select "`Properties`"
- Select the "`Web Sharing`" Tab

ABB. All rights reserved

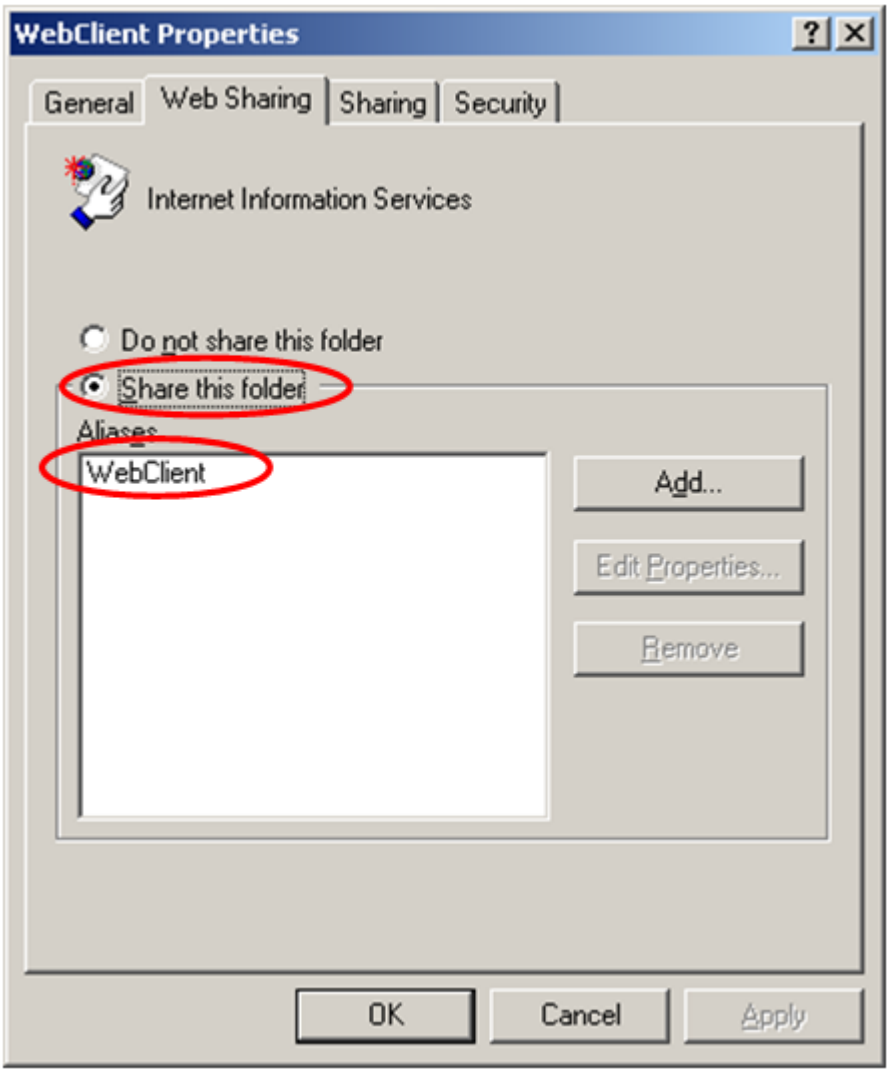


Figure 168 – Web Sharing

Select “Share this folder”
Select “WebClient” from the Aliases list and click on “Edit Properties”, in the opened windows select checkbox as image:

ABB. All rights reserved

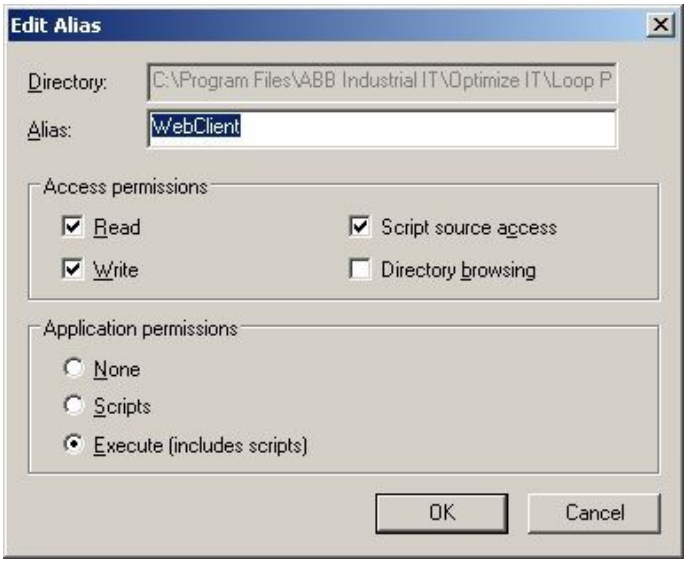


Figure 169 – Web folder configuration

Click the "OK"

At this point, if Web Client is correctly licensed, is it possible to open it with using MS Internet Explorer and entering typing in the address bar "http:///*[SERVER-NAME]*/WebClient" (Where *[SERVER-NAME]* is the network name/address of the LPM server PC)

In order to increase security, suggested first step to take is to change the default login credentials for Web client users (defaults: User: admin, Password; admin, User: adm, Password: adm, and User: usr, Password: usr,) and create and configure permissions for all the needed user.
Before finish the above explained operation, ensure that a user with Administrator privileges always exists.

ABB. All rights reserved

1. Additional Settings for "WebClient" virtual directory:

A) Go to Properties of **Virtual Directory** and click **Configuration** button.

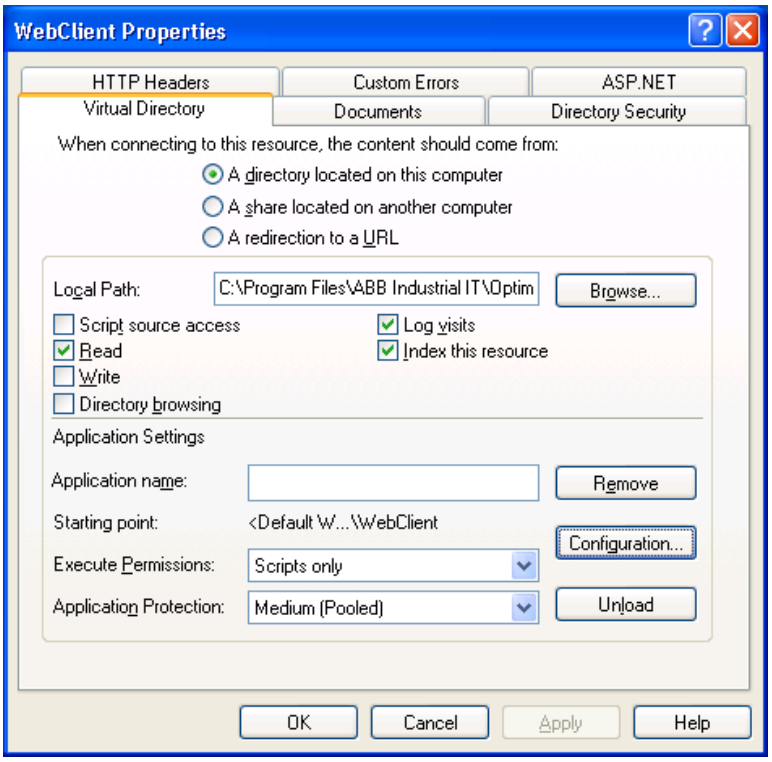


Figure 170 – WebClient Properties – Virtual Directory Tab

ABB. All rights reserved

B) Ensure “Enable parent paths” is enabled.

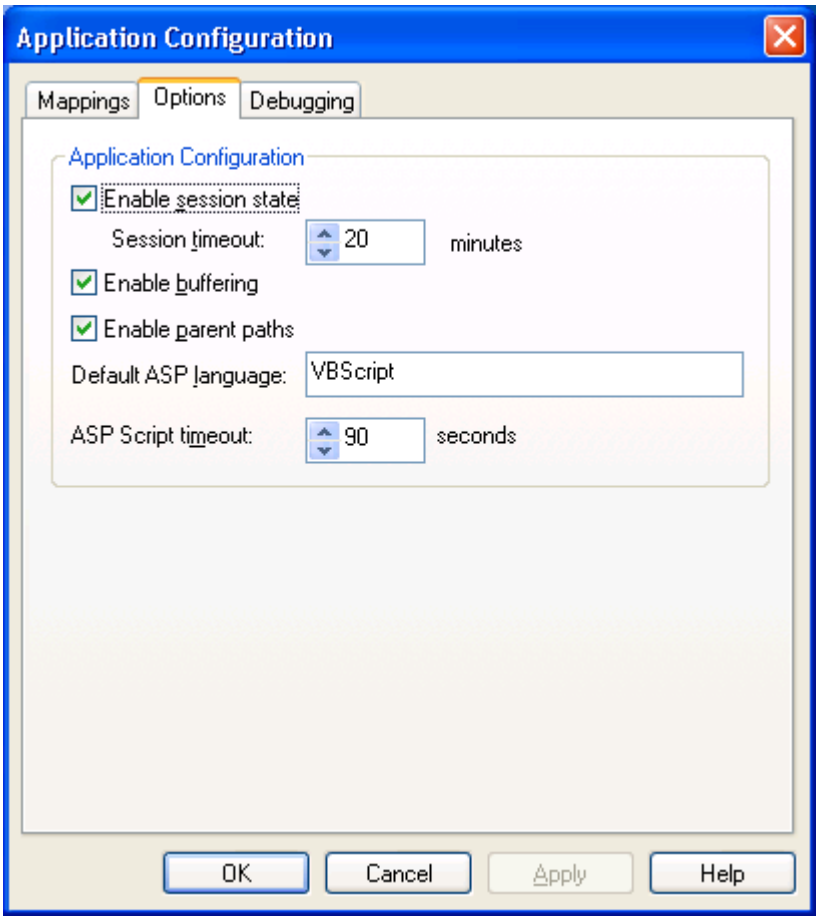


Figure 171 – Application Configuration – Options Tab

ABB. All rights reserved

2. Read/Write permission for IWAM_machinename user on “WebClient” folder

A) Right click on "WebClient" folder under LPM Install Directory, Go to Properties and click **Security** tab. Click “Add” button.

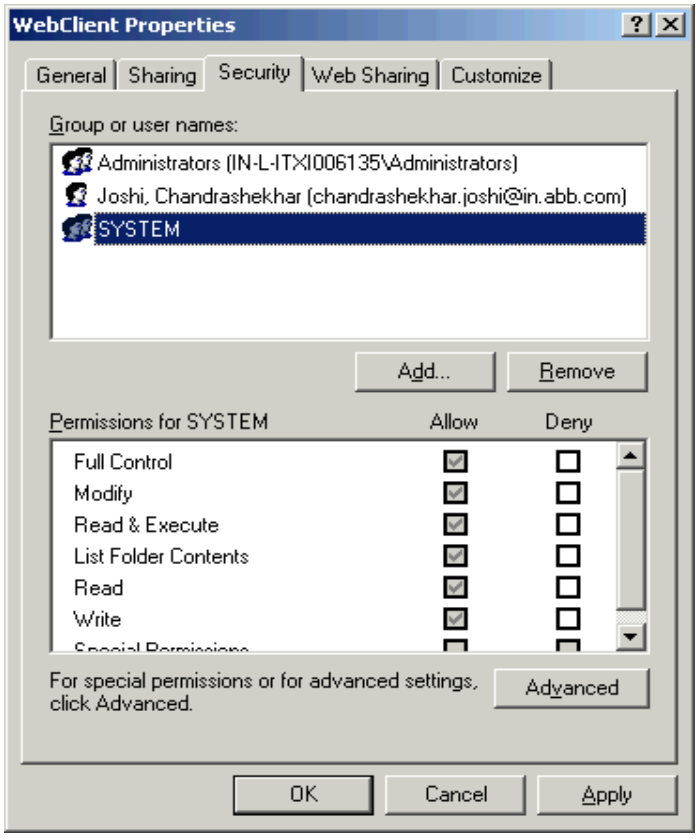


Figure 172 – WebClient Properties – Security Tab

ABB. All rights reserved

B) Find all users in local machine and select IWAM_machinename user

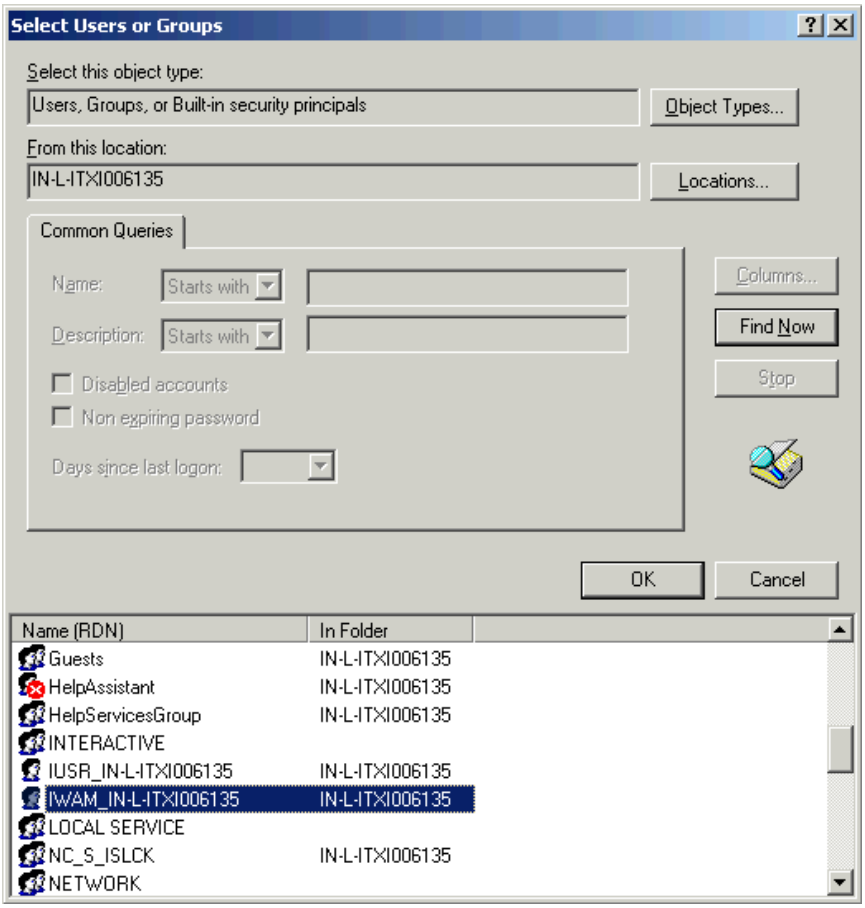


Figure 173 – Select Users or Groups

ABB. All rights reserved

C) Allow Write permission to this user for particular folder

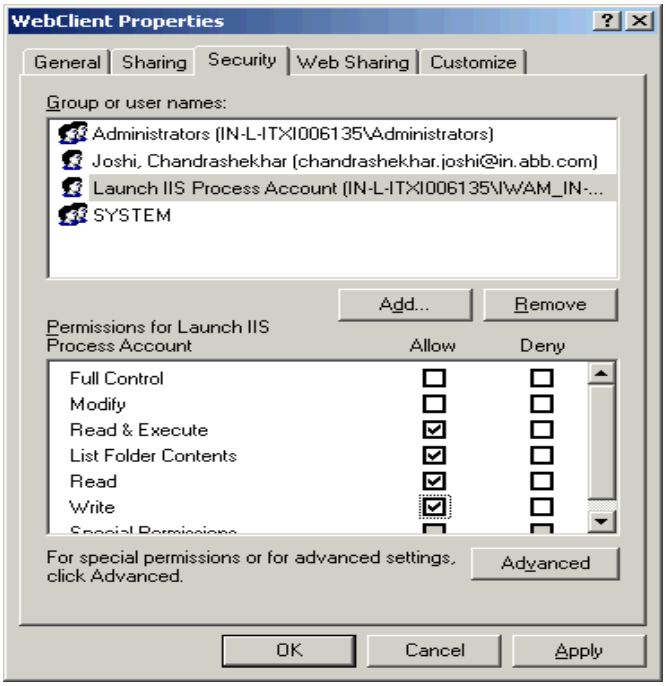


Figure 174 – WebClient Properties – Security Tab

ABB. All rights reserved

7. LPM Interface with KM 6.1

7.1 *Overview*

Owners of large plants are confronted with the need to maintain thousands of loops distributed over a large geographical area, and probably over several control networks. Under this circumstance it is necessary to provide robust data acquisition procedures, including for instance local data collector nodes, and also easy to access reporting capabilities via the site intranet.

To address this need ABB has created software components that allow to connect LPM to its flagship historian cpmPlus History / Knowledge Manager 6.1. This allows use these databases as data sources for auditing and tuning, and as data repositories of all indexes and diagnostics. In addition, reports via Web

The cpmPlus LPM version 3.3 features extension of cpmPlus KM by offering following distinct futures:

- KM as additional tuning/auditing data source. (in addition to existing OPC/Infi90 support)
- KM as database for storing the KPI(Indices and Diagnostics), and
- Generic and custom reports types for KPI analysis in KM Personal Assistant.

7.2 *Scope and requirements*

This KM feature extension is optional, and will work only if you have an existing KM server setup. Currently the cpmPlus LPM V 3.3 supports this future for KM 6.1 version only.

You have an option to use or disable this future. To turn ON/OFF this facility is explained in the below sections later.

If you intend to use this future it is recommended KM 6.1 is installed on a different server machine. This feature also requires you to install all the patches provided with KM 6.1 release in addition to “KM6.1 LPM extensions” patch. This patch extends the support for communication between LPM and KM database.

For the sake of the KM6.1 functionality being explained here, installations of cpmPlus Expert Optimizer 6.1 would also provide the desired behaviour.

ABB. All rights reserved

7.3 *Configuring the KM Server*

Once a KM Server is installed and configured with “KM6.1 LPM extensions”, its time for LPM to know about the details of this KM server. In LPM client, click on Tools ,Options. Under the tab “KM server” configure the requested details and press OK button. Such typical window is shown in Figure 175. Connecting KM 6.1 server to LPM 3.3.

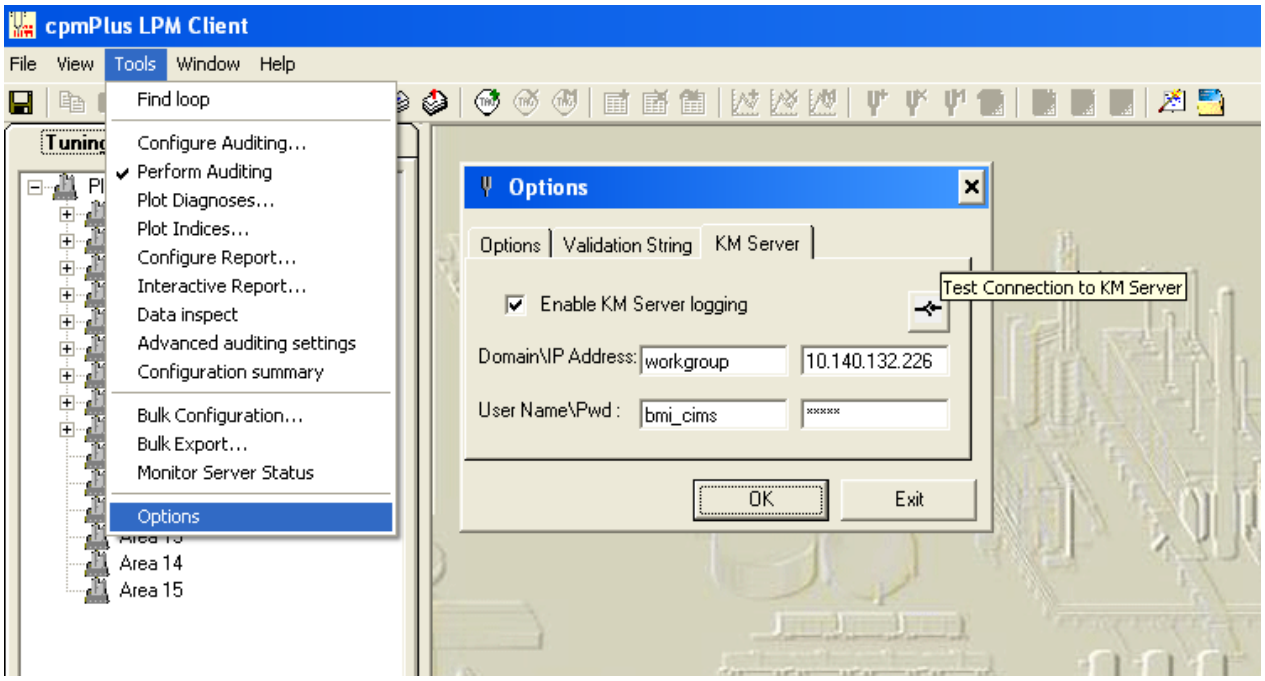


Figure 175. Connecting KM 6.1 server to LPM 3.3

7.3.1 *Test Connection to KM Server*

To test the connection to KM Server, then you need to re-open the Options, and click on the button provided. If all the KM server logon information is correct and the server is reachable, then it will pop-up a message appropriately.

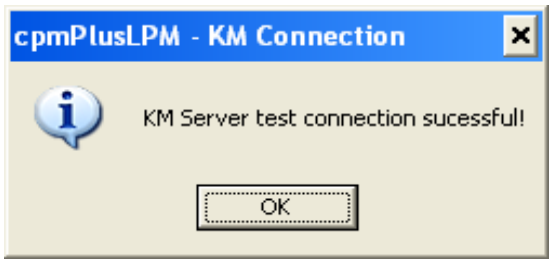


Figure 176. Testing KM 6.1 connection

ABB. All rights reserved

7.4 KM as Tuning data source

Once the KM server configuration is complete as above, you can start configuring areas that would use KM as data source.

7.4.1 Configure KM Area

It is required to configure a separate area dedicated to this KM related loops. As an example,, once such area configuration is as shown in the *Figure 177. Configuring Area with KM 6.1 as data source.* below. When you select the data source in the combo box as "Knowledge Manager" The IP Address is automatically selected as configured before. Fill in the other required details and proceed.

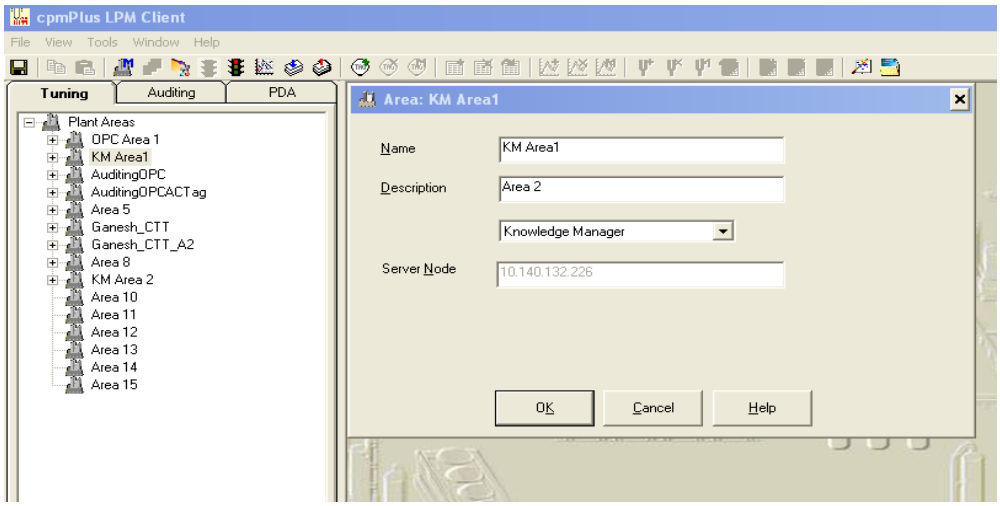


Figure 177. Configuring Area with KM 6.1 as data source.

7.4.2 Configure KM Loop

Once the above step is done, you can add and configure a new loop. Unlike configuring the signals directly, in case of KM, it is required to configure the log names as defined in KM. This is because, it is from these KM logs, and the data is read by LPM. Below is the example how KM Logs are configured.

ABB. All rights reserved

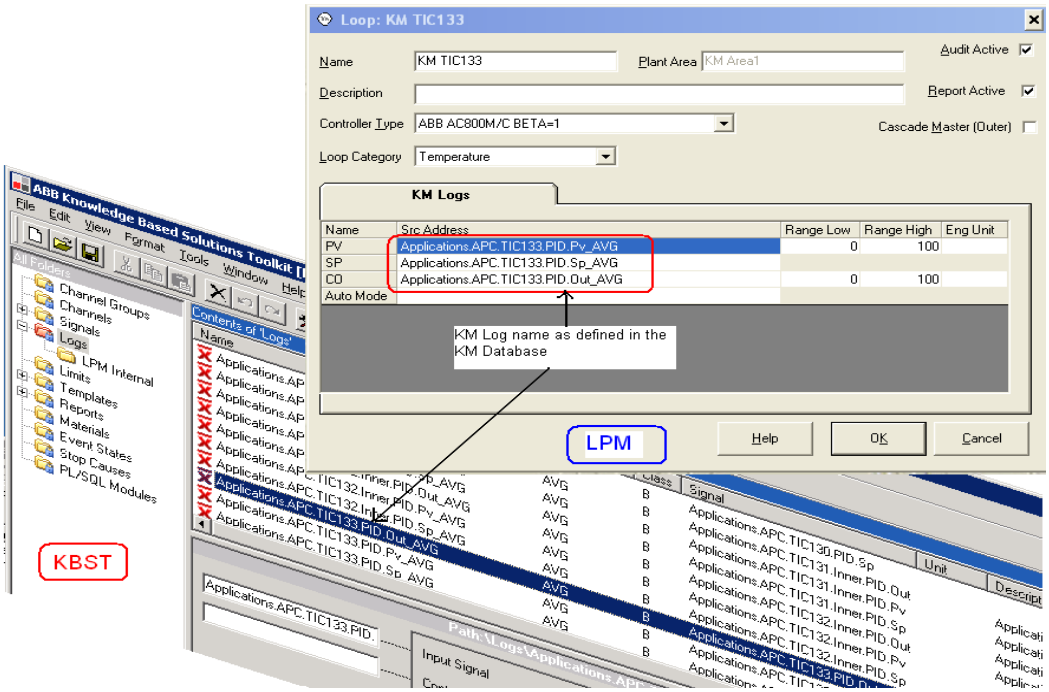


Figure 178. Loop configuration with KM 6.1 as data source

Please note that if you do not enter correct tag name, LPM may treat the tag as non-existing, and will generate an appropriate message.

7.4.3 Data Collection

Once the tags are properly configured, you can start a new data collection to check. See below *Figure 179. Data collection for tuning with KM 6.1 as data source* how to enter the necessary details for KM data collection.

ABB. All rights reserved

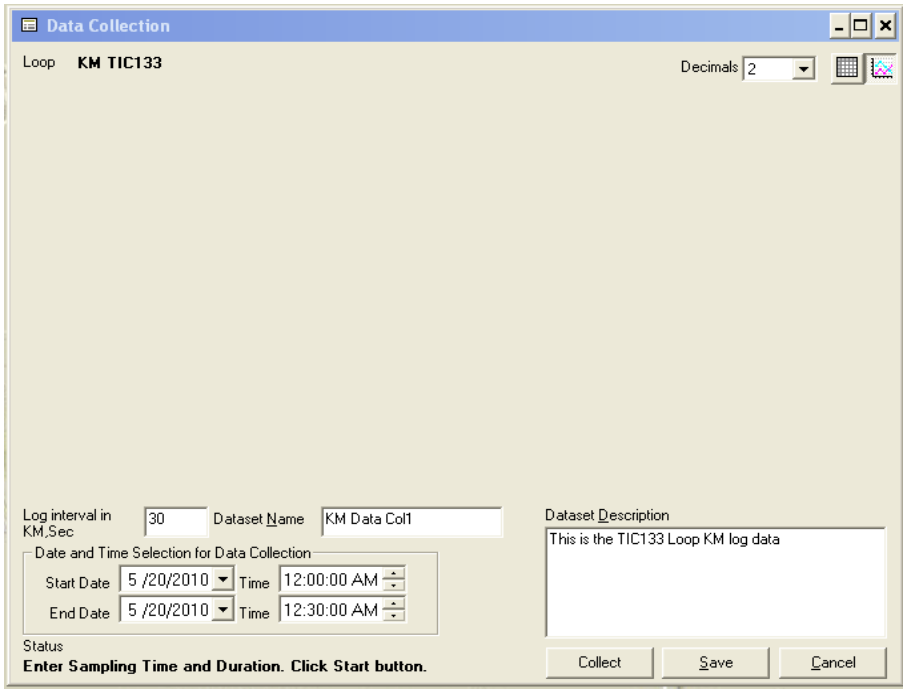


Figure 179. Data collection for tuning with KM 6.1 as data source

Enter the sampling rate as configured in KM database for the signals. You can select the start and end date for data collection. And press on “Collect” button.

If the Log interval in KM (see above figure), is different than actual interval in KM, then you will see a question prompted by LPM as below:

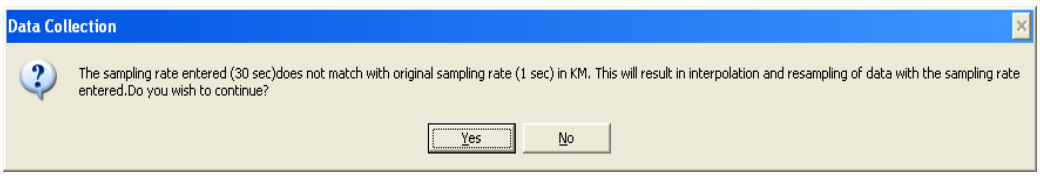

Data Collection
The sampling rate entered (30 sec)does not match with original sampling rate (1 sec) in KM. This will result in interpolation and resampling of data with the sampling rate entered.Do you wish to continue?
Yes
No

Figure 180. Data Collection Option Dialog

ABB. All rights reserved

If you choose “Yes” then LPM starts data collection and you will see the status like:

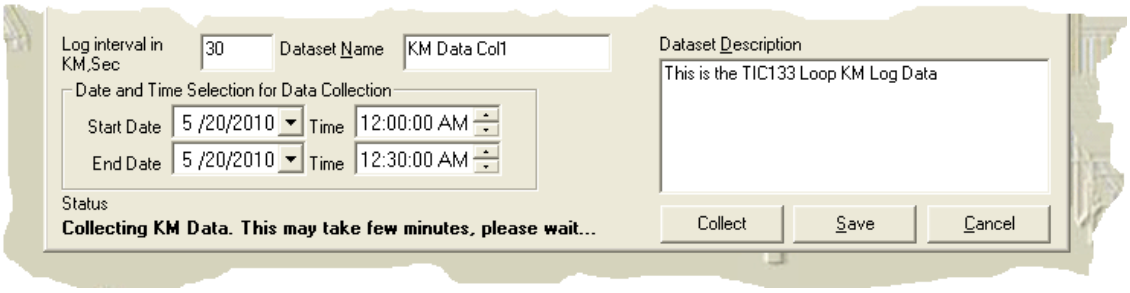


Figure 181. Data Collection – Dataset Description

Once the data collection is complete, you will get the status as below:

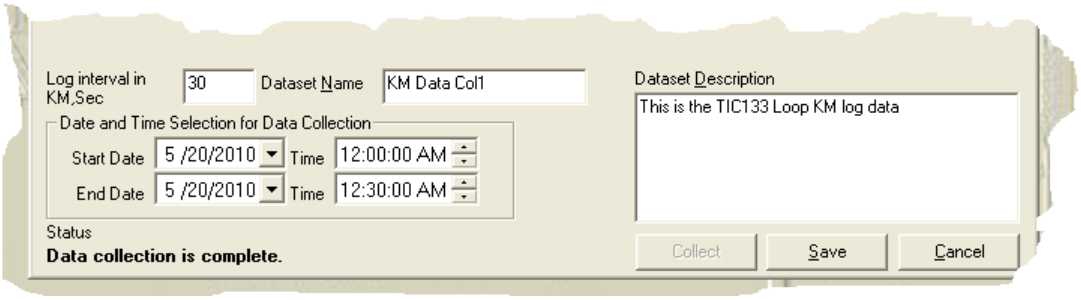


Figure 182. Data Collection - Complete

Once you save the collection, close the Data Collection window, and re-open to see the saved collection. You can then proceed with steps for Modelling and Tuning as defined in the User Manual.

ABB. All rights reserved

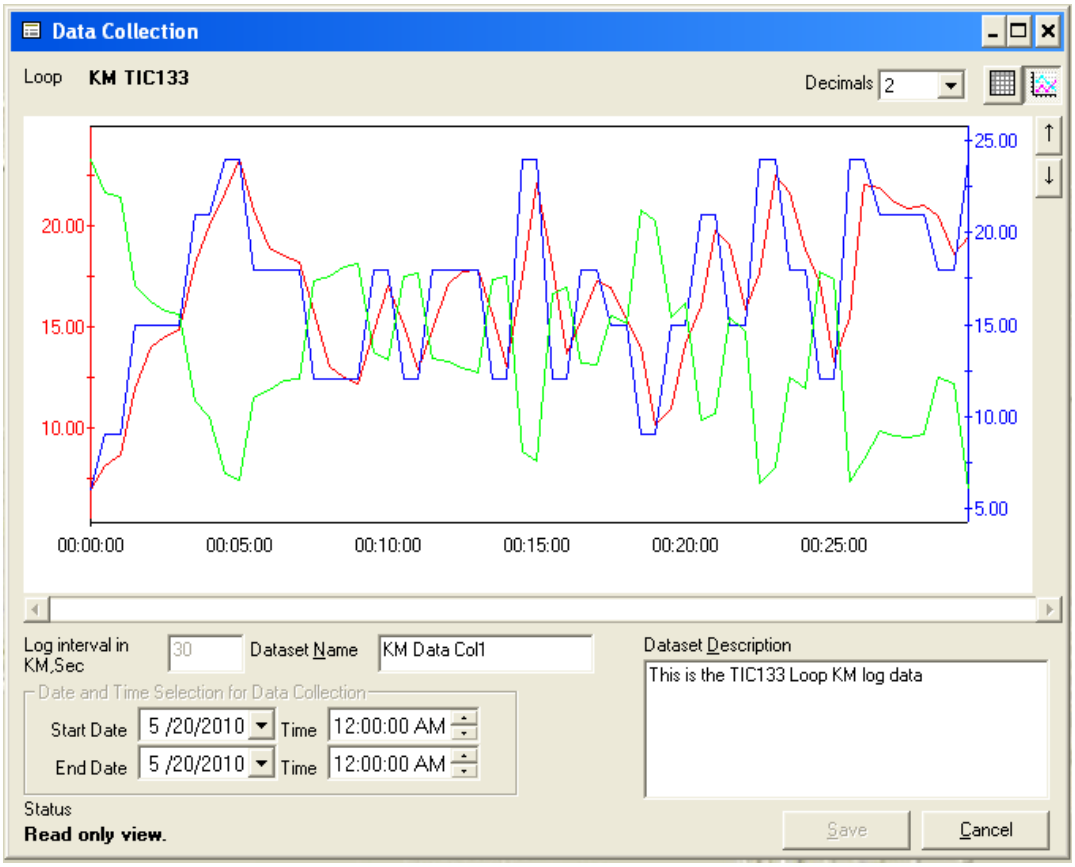


Figure 183. Data collected from KM 6.1 logs

7.5 *KM as Auditing data source*

7.5.1 Configuration

Once the KM Area Loops are configured with correct log names, and when the "Auditing Active" check box is selected, the auditing process will begin as per the Auditing Configuration selected. In case the data source for an auditing collection is the OPC/Infi90, then the samples are read at every "Sampling rate" seconds. However, in case of KM, actual batch/continuous data collection happens at end of the collection period.

7.5.2 Raw audit data collection

KM audit read data happens at end of the collection period. In order to give the user feedback on the status of the collection process and of the quality of the data being collected, raw audit data collection is performed every 5mins. This can be seen from "View Auditing Data" for any loop that is configured for auditing. It can be seen in below that for the batch and continuous collections, for KM TIC 133 loop, the raw data collection is happening at every 5 minutes. It should be noted in the below *Figure 184. Raw audit data collection* that the continuous collection is set at 30sec as sampling rate and the batch collection is set to 3 sec.

ABB. All rights reserved

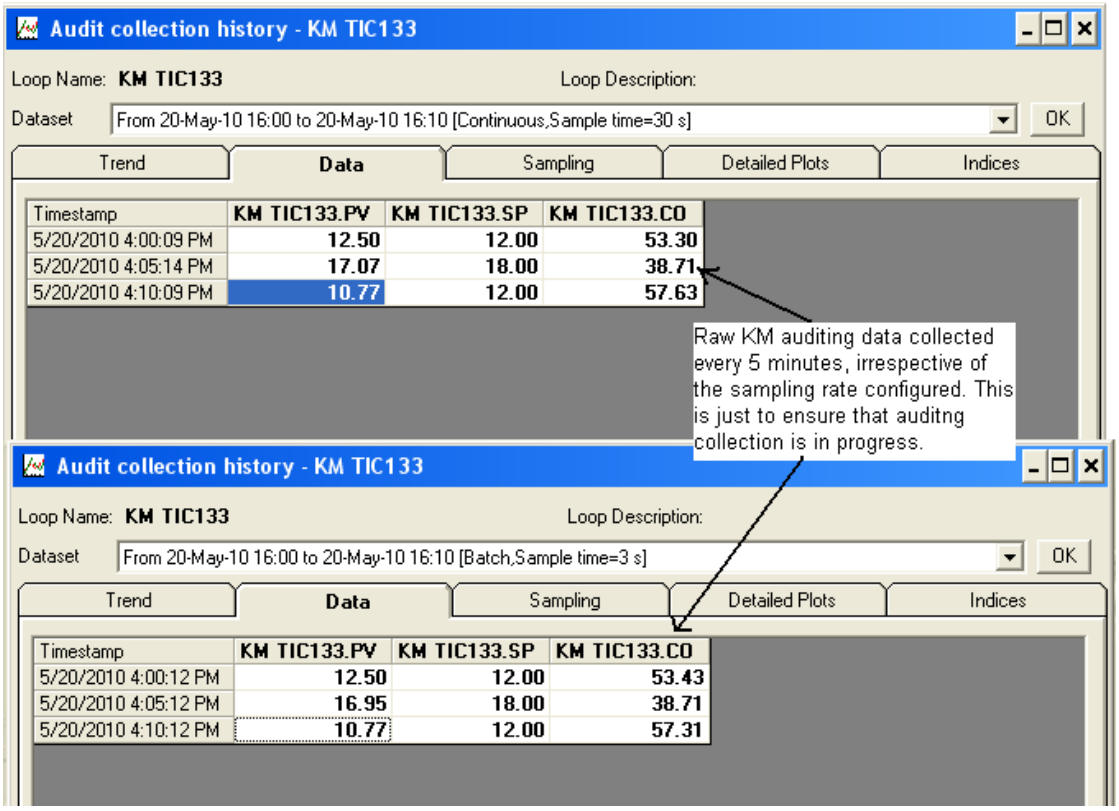


Figure 184. Raw audit data collection

7.5.3 Audit data collection

At the end of the collection period, the auditing server reads the actual KM log data for loop parameters from KM for the entire collection period at once. The KM Logs read from KM may be at a sampling rate that is different from the rate that is configured in LPM. Thus there might be a mismatch of sampling frequency between the KM logs and what LPM requires. To fill this gap, LPM reads all the required logs at the PRIMARY log rate in KM, and resamples the KM data, to get the required sampling rate. Once this is done, the Raw audit data collection is overwritten by this re-sampled data. You can find from the below *Figure 185 Final stage of audit data collection from KM.* that the raw audit data is replaced at the end of the collection period.

ABB. All rights reserved

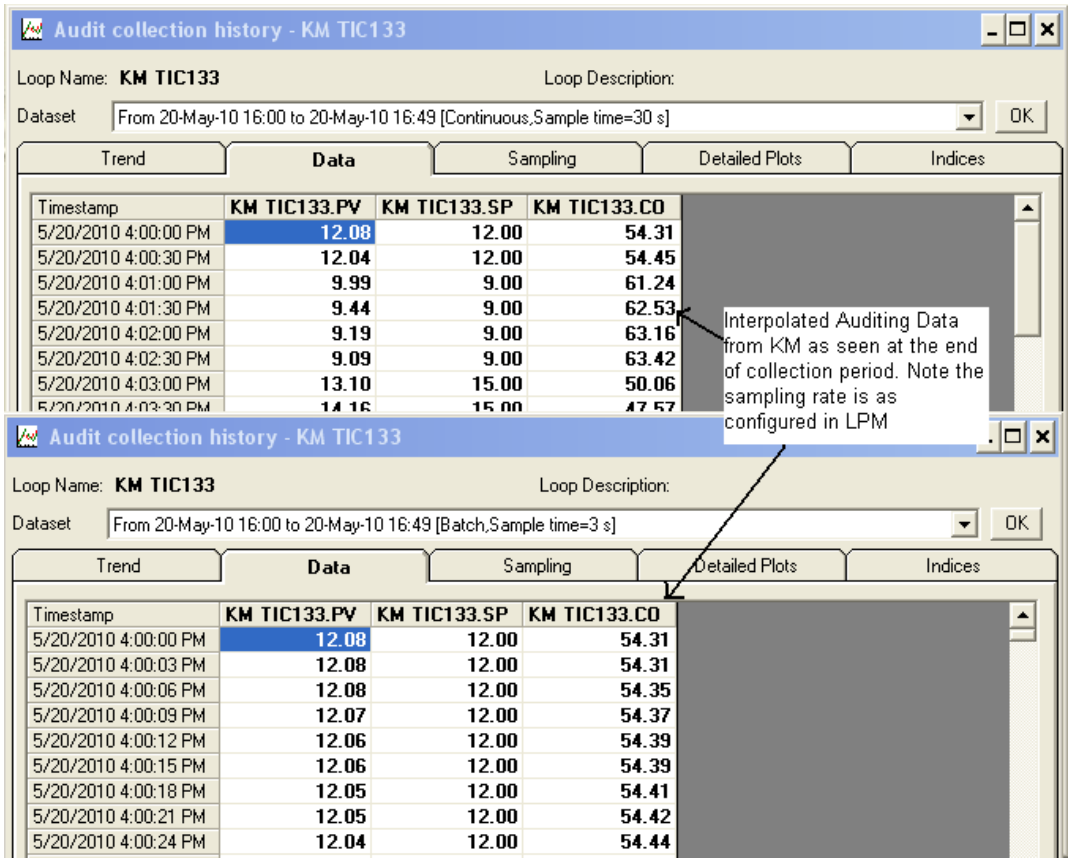

Audit collection history - KM TIC133
Loop Name: KM TIC133
Loop Description:
Dataset
From 20-May-10 16:00 to 20-May-10 16:49 [Continuous,Sample time=30 s]
OK
Trend
Data
Sampling
Detailed Plots
Indices
Timestamp
KM TIC133.PV
KM TIC133.SP
KM TIC133.CO
5/20/2010 4:00:00 PM 12.08 12.00 54.31
5/20/2010 4:00:30 PM 12.04 12.00 54.45
5/20/2010 4:01:00 PM 9.99 9.00 61.24
5/20/2010 4:01:30 PM 9.44 9.00 62.53
5/20/2010 4:02:00 PM 9.19 9.00 63.16
5/20/2010 4:02:30 PM 9.09 9.00 63.42
5/20/2010 4:03:00 PM 13.10 15.00 50.06
Interpolated Auditing Data from KM as seen at the end of collection period. Note the sampling rate is as configured in LPM
Audit collection history - KM TIC133
Loop Name: KM TIC133
Loop Description:
Dataset
From 20-May-10 16:00 to 20-May-10 16:49 [Batch,Sample time=3 s]
OK
Trend
Data
Sampling
Detailed Plots
Indices
Timestamp
KM TIC133.PV
KM TIC133.SP
KM TIC133.CO
5/20/2010 4:00:00 PM 12.08 12.00 54.31
5/20/2010 4:00:03 PM 12.08 12.00 54.31
5/20/2010 4:00:06 PM 12.08 12.00 54.35
5/20/2010 4:00:09 PM 12.07 12.00 54.37
5/20/2010 4:00:12 PM 12.06 12.00 54.39
5/20/2010 4:00:15 PM 12.06 12.00 54.39
5/20/2010 4:00:18 PM 12.05 12.00 54.41
5/20/2010 4:00:21 PM 12.05 12.00 54.42
5/20/2010 4:00:24 PM 12.04 12.00 54.44

Figure 185 Final stage of audit data collection from KM.

7.6 *KM as KPI Database*

Once the KM Server is configured and auditing is enabled, the KPIs (Indices and Diagnostics) are written to KM database. These KPIs are then available for viewing in the KM database. Note that the KPI storage happens for all the loops irrespective of the data source type. This means that once you enable "KM Server logging" under the Options dialogue, see *Figure 175. Connecting KM 6.1 server to LPM 3.3*, the calculated KPIs are stored in KM for all loops (OPC/Infi90/Knowledge Manager).

To verify that the KPIs are being logged successfully into KM, you could use KBST tool in the KM Server. When you browse to the Signals view you should be able to see LPM Internal signals and Logs configured automatically like shown below.

ABB. All rights reserved

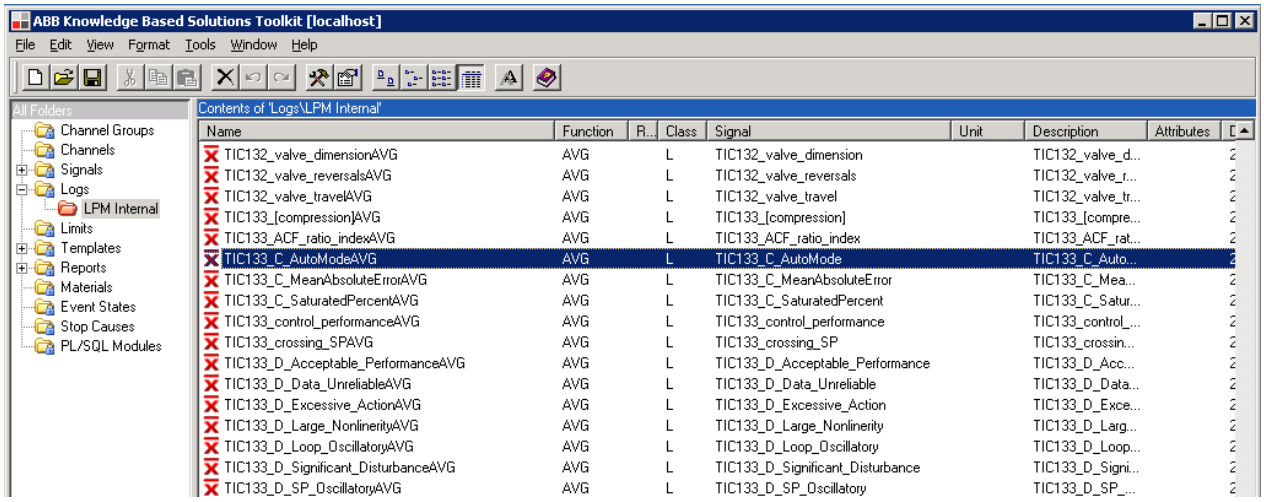


Figure 186. KPIs and Diagnostic in KM 6.1 Database

7.7 *LPM reports in KM Personal Assistant*

7.7.1 Standard LPM reports in KM

KM will provide some basic standard reports for displaying LPM data. These should require little or no configuration once imported in KM.

7.7.2 Importing Standard Reports in KM

The reports are distributed as an xml file. This is a standard KM xml "export" file. Use the KBST to import the reports and templates. (Tools – Import). Select the file that is supplied in the CD along with the LPM installer.

7.7.3 Viewing the LPM reports in the KM personal Assistant

Once the reports are imported they should be visible in the report tree:

Figure 187. KM Audit Reports

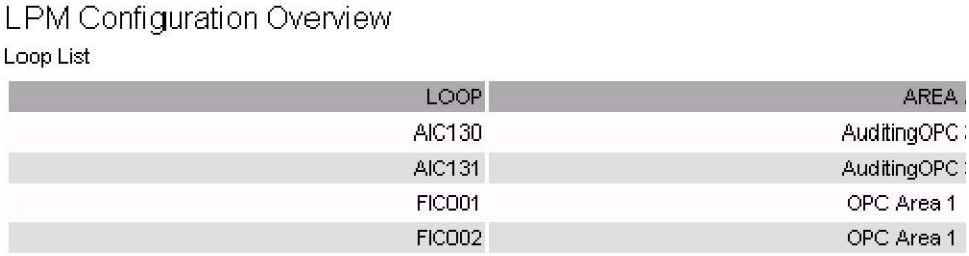
ABB. All rights reserved

7.7.4 System Reports

The system reports are in a sub folder called “system”. These are intended for use by the LPM developer/engineer who would require viewing general information about the LPM data stored in KM.

7.7.4.1 Overview Report

This report contains two sections. The upper section shows a list of all the Loops that have been written into KM from LPM.

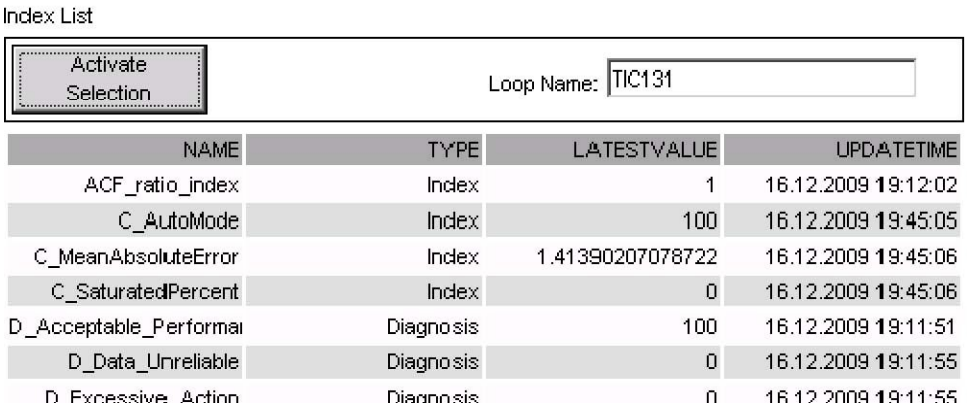

LPM Configuration Overview

Loop List

LOOP	AREA
AIC130	AuditingOPC
AIC131	AuditingOPC
FIC001	OPC Area 1
FIC002	OPC Area 1

Figure 188. Report Overview

The second section allows the user to drill down to index information about a specific loop. Type the loop name (or copy + paste) into the Loop Name filter box, and click the Activate Selection button. This will show a sub table containing details of the Index Name, Index Type, Last Value Received and the time the last value was received.


Index List

Activate Selection

Loop Name: TIC131

NAME	TYPE	LATESTVALUE	UPDATETIME
ACF_ratio_index	Index	1	16.12.2009 19:12:02
C_AutoMode	Index	100	16.12.2009 19:45:05
C_MeanAbsoluteError	Index	1.41390207078722	16.12.2009 19:45:06
C_SaturatedPercent	Index	0	16.12.2009 19:45:06
D_Acceptable_Performai	Diagnosis	100	16.12.2009 19:11:51
D_Data_Unreliable	Diagnosis	0	16.12.2009 19:11:55
D_Excessive_Action	Diagnosis	0	16.12.2009 19:11:55

Figure 189. Indexes in KM 6.1 Reports

7.7.4.2 Inactive Indices Report

This report shows a list of the Indexes that are stored in KM, but have not been updated for at least 24 hours. This may be useful to show the LPM engineer which loops can be deleted or to highlight a configuration problem.

ABB. All rights reserved

7.7.5 User Reports

The user reports are in a sub folder “user”. These reports are intended to show tabular historical data about the loop auditing indexes. In all these reports the user must specify a time range and a “period type”. When displaying a period type chooses “Primary Log Value” to see a report based on the exact “raw” data that was written from LPM. Choose Hourly, Daily, Monthly etc. to see a report based on the KM consolidated log data (e.g. Average) for the specific period type. This KM consolidated data is created in the data base by a background process and will give faster access to reports. Certain reports will show all the data records between the time range (e.g. the Loop Audit Data) whereas other reports will show an average value of all data points between the time range. Certain reports show a “quality” value. This is the percentage of time in the consolidation period that the log (index) had a valid value.

7.7.5.1 Loop Audit Data and Loop Audit Data Averaged

From this report the user chooses a time range, period type and loop in the header, and clicks “Activate Selection”. This report then shows a table with the index names, timestamp, values and quality value. This report does not aggregate the data on the fly, it shows an exact one-to-one of what is stored in the KM database. This is why there can be more than one row per index.

Figure 190. Loop Audit Data

This report is the same as the Loop Audit Data report, except that each index shows only one row. To do this the values and quality values are averaged at user request. The report also shows the number of records that were averaged.

ABB. All rights reserved

Figure 191. Loop Data Averaged

7.7.5.2 Loop Audit Diagnostics

This report shows the loop data in the rows and the diagnostics in the columns. The user can choose the time range, period and an optional area name filter. Note that the values here are averaged over the given time period to enable only one value to be displayed per loop/diagnosis.

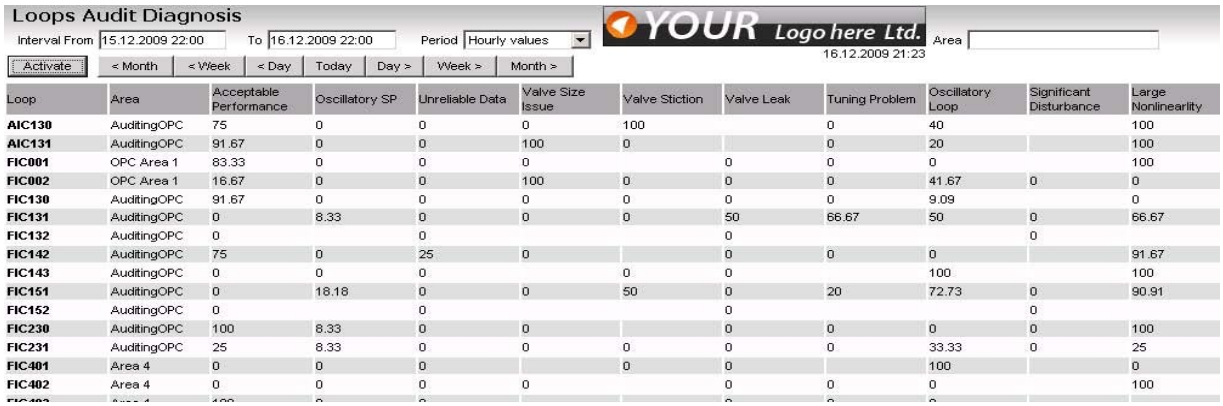

Loops Audit Diagnosis
Interval From 15.12.2009 22:00
To 16.12.2009 22:00
Period Hourly values
YOUR Logo here Ltd.
16.12.2009 21:23
Area
Activate
< Month
< Week
< Day
Today
Day >
Week >
Month >
Loop
Area
Acceptable Performance
Oscillatory SP
Unreliable Data
Valve Size Issue
Valve Stiction
Valve Leak
Tuning Problem
Oscillatory Loop
Significant Disturbance
Large Nonlinearity
AIC130 AuditingOPC 75 0 0 0 100 0 40 100
AIC131 AuditingOPC 91.67 0 0 100 0 0 20 100
FIC001 OPC Area 1 83.33 0 0 0 0 0 0 100
FIC002 OPC Area 1 16.67 0 0 100 0 0 0 41.67 0 0
FIC130 AuditingOPC 91.67 0 0 0 0 0 0 9.09 0
FIC131 AuditingOPC 0 8.33 0 0 0 50 66.67 50 0 66.67
FIC132 AuditingOPC 0 0 0 0 0
FIC142 AuditingOPC 75 0 25 0 0 0 0 91.67
FIC143 AuditingOPC 0 0 0 0 0 100 100
FIC151 AuditingOPC 0 18.18 0 0 50 0 20 72.73 0 90.91
FIC152 AuditingOPC 0 0 0 0 0
FIC230 AuditingOPC 100 8.33 0 0 0 0 0 0 100
FIC231 AuditingOPC 25 8.33 0 0 0 0 0 33.33 0 25
FIC401 Area 4 0 0 0 0 0 100 0
FIC402 Area 4 0 0 0 0 0 0 0 100

Figure 192. Diagnostic in KM 6.1

7.7.5.3 Loop Audit Preconditions

This is the same as the Loops Audit Diagnosis report except that it displays Preconditions.

7.7.5.4 Loop Audit Diagnostics Best Worst – Worst Best

This is similar to the Loops Audit Diagnosis report. In this case the user gives a Diagnosis name in the header. The report then shows the Loop Names ordered by the value of the Diagnosis. This ordering is from low to high.

The same as the previous report but with the ordering is from high to low also exists.

ABB. All rights reserved

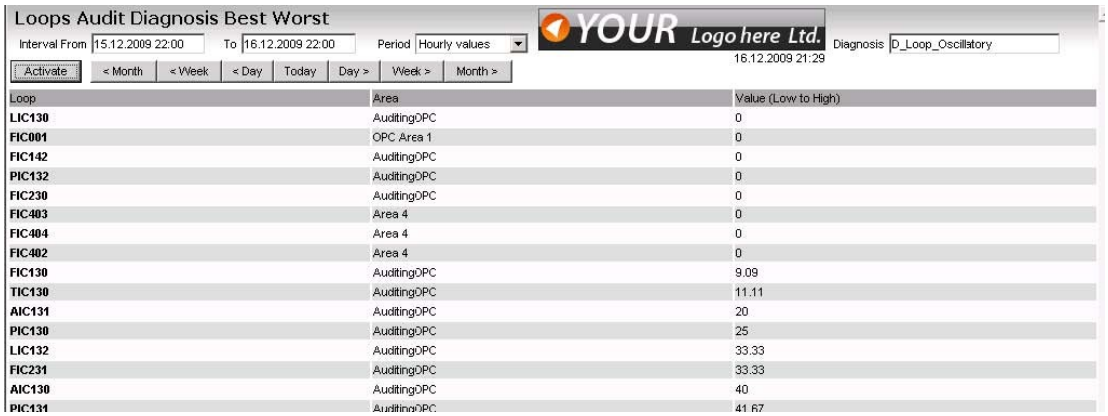


Figure 193. Worst / Best Loops Report

ABB. All rights reserved

8. Utilities

8.1 *Loop find*

The loop find feature can be activated by selecting the Tools/Loop find entry in the menu. The feature allows to quickly search a certain loopname in the LPM configuration. This feature is quite useful in case of large cases, where loop search in the configuration tree is not trivial.

Loop find allows to define a loopname filter that is activated by pressing the update button. All loop names starting with the selected filter characters will be presented in the loops list.

By selecting one of the loops listed and clicking the Show pushbutton, the user can highlight the selected loop in the LPM configuration tree. By selecting Open, the user can open the configuration form for the selected loop. Cancel allows to close the form.

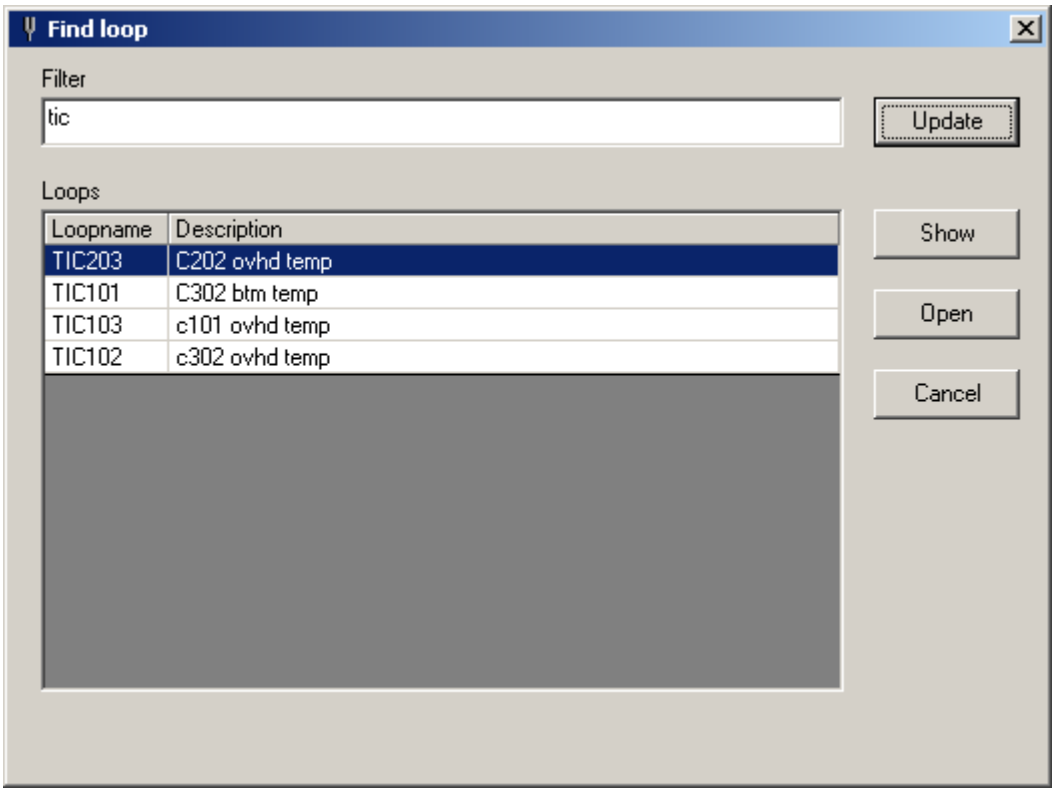


Figure 194 – Loop find

8.2 *Configuration Summary*

The LPM Configuration Summary feature allows the user to have a global overview of the configuration. Configuration data is organized into several tabs, detailed in the next paragraphs.

ABB. All rights reserved

8.2.1 General tab

This tab presents general information on the configuration. The table below presents information showed in the main grid.

Description	Notes
Number of loops configured	
Number of loops active for auditing	
Number of loops active for reports	
Number of loops licensed (auditing license)	
Number of loops still available	computed as number of loops licensed – number of loops currently active for auditing
Number of areas	
Number of areas used	Areas where loops are configured
Number of categories	
Number of categories used	Categories where loops are configured
Number of reports configured	
Number of datasets	Tuning datasets objects
Number of models	Model objects
Number of tuning	Tuning objects
Number of logs	Tuning logs

Table 11 – General information on configuration

The form features a pie-chart presenting auditing license usage – used versus still available loops and a histogram charts summarizing object counts. The next *Figure 195 – General tab* shows the general appearance of the form.

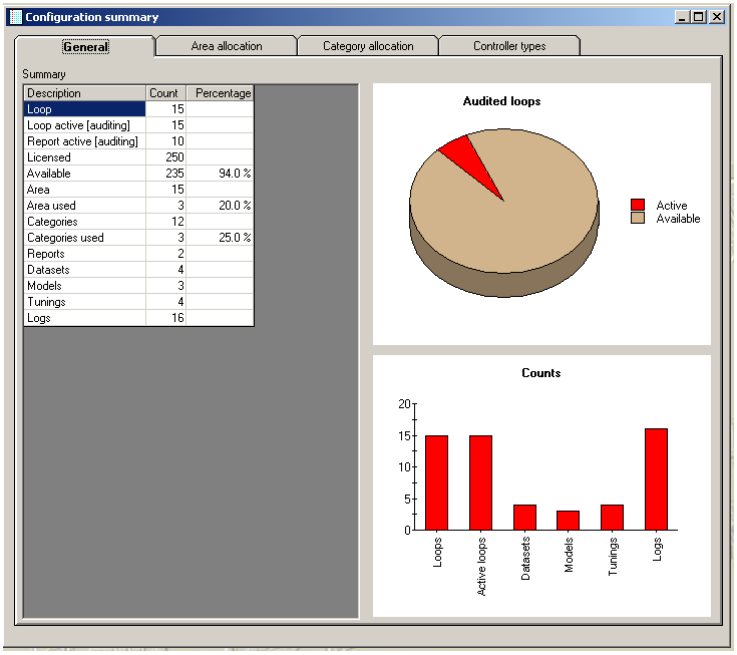
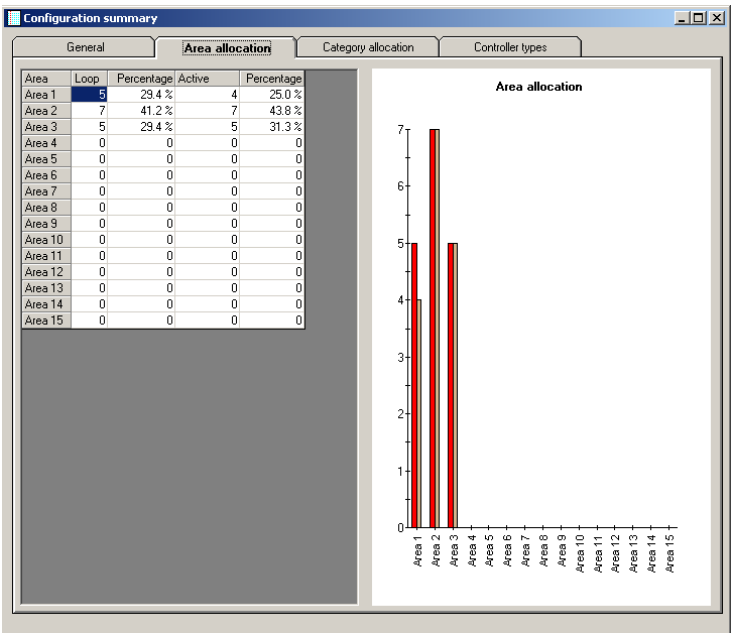


Figure 195 – General tab

ABB. All rights reserved

8.2.2 Area Allocation

The Area Allocation tab presents loop allocation with respect to areas. Both the configured and auditing active loops are presented in the form grid and histogram chart. The next picture shows a sample case.

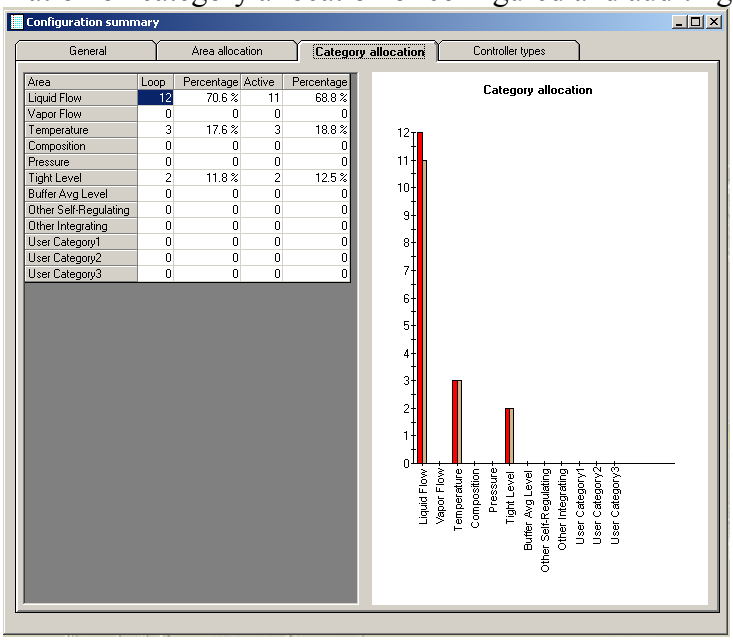


Area	Loop	Percentage	Active	Percentage
Area 1	5	29.4 %	4	25.0 %
Area 2	7	41.2 %	7	43.8 %
Area 3	5	29.4 %	5	31.3 %
Area 4	0	0	0	0
Area 5	0	0	0	0
Area 6	0	0	0	0
Area 7	0	0	0	0
Area 8	0	0	0	0
Area 9	0	0	0	0
Area 10	0	0	0	0
Area 11	0	0	0	0
Area 12	0	0	0	0
Area 13	0	0	0	0
Area 14	0	0	0	0
Area 15	0	0	0	0

Figure 196 – Area allocation tab

8.2.3 Category Allocation Tab

This tab presents information on category allocation of configured and auditing active loops.



Area	Loop	Percentage	Active	Percentage
Liquid Flow	12	70.6 %	11	68.8 %
Vapor Flow	0	0	0	0
Temperature	3	17.6 %	3	18.8 %
Composition	0	0	0	0
Pressure	0	0	0	0
Tight Level	2	11.8 %	2	12.5 %
Buffer Avg Level	0	0	0	0
Other Self-Regulating	0	0	0	0
Other Integrating	0	0	0	0
User Category1	0	0	0	0
User Category2	0	0	0	0
User Category3	0	0	0	0

Figure 197 – Category allocation tab

ABB. All rights reserved

8.2.4 Controller type

This tab presents information on the different controller types. The different types are categorized based on their algorithm.

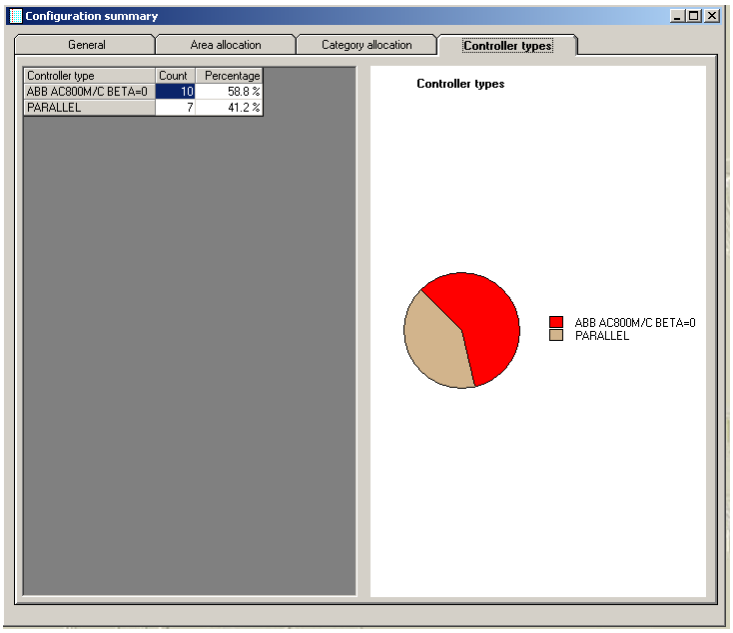


Figure 198 – Controller allocation tab

8.3 Options

This feature allows the user to define LPM specific options. The feature can be accessed by pressing Tool/Options menu entry. A dedicated form is presented to allow the user selecting LPM parameters. The form has two separate tabs. While the first tab is devoted to defining specific LPM parameters, the second tab is reserved to entry/revision on LPM validation string.

The next figure presents the form first tab:

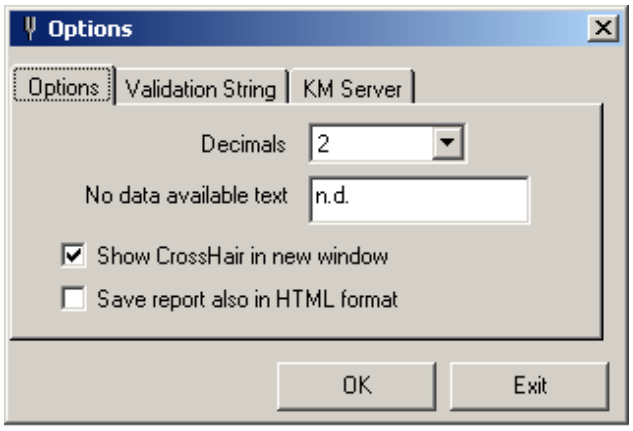


Figure 199 – Option form, tab 1

ABB. All rights reserved

The parameters allow customizing LPM behavior as follows:

Item	Description	Notes
Decimals	Allows to customize the number of decimals used for data presentation	
No data available text	Allows customizing text data used in case of inconclusive indices or no index values available in the interactive reports. As an example, if stiction for a given loop cannot be detected nor ruled out, this text will be presented in the reports.	
Show cross hair in new window	Allows selecting whether to show cross hair values in a new window or in the existing window. The first option	
Save report also in HTML format	Allows selecting whether to save LPM reports in HTML format (in addition to the Microsoft Excel format).	

Table 12 – Options, first tab

ABB. All rights reserved

The second tab presents the validation string. Modifications to the validation string can be applied by pressing the change pushbutton.

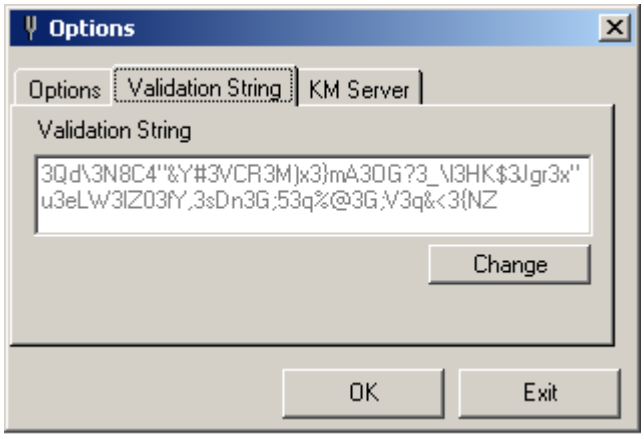


Figure 200 – Option form, Tab 2

The third tab presents the "KM Server" details. Please refer to figure 175. If you want to use KM 6.1 in LPM, you should click on the check box "Enable KM Server Logging". If you do not use KM6.1, you should not use this tab, and keep the check box unchecked. When you select this check box, the credential fields like: user name/password will appear. It is compulsory to enter all correct information here so that the LPM-KM interfacing to work. For more information refer to section 7.3.

ABB. All rights reserved

9. Appendix A - Chart Operating Instructions

On relevant dataset windows, the user can view the trends of datasets by selecting the **Trend** button. For instance, the Trend button appears on the Tags tab for the raw dataset and also on the *(1) Select Tags* tab for the child dataset. Dataset trends are also available when performing specific dataset operations (that is, excluding slices of data).
The following operations are available with the Trends.

- *Number of Charts to Display*. Selection box for the user to select the number of charts to display. Additional charts can be user defined.
- *Zoom In*. Button to zoom in on the Trend (horizontal scroll bar permits viewing any portion of the trend).
- *Zoom Out*. Button to zoom out of the Trend.
- *Zoom Select*. Permits selection of a specific area of the trend for zooming in.
- *Mark Slice*. Permits selection of specific slice of trend data. (Used for excluding bad data).

In addition, there are several right click menu options that are described in Table 13.

Item	Description
Grids	Option to display X grids, Y grids, or both.
Cross Hair	Show specific values of points on plot area via cross hair.
Show Timestamps	Option to show X-axis as either timestamp values or sample number.
Setup Trends	Used to configure trend displays.
Resize X Axes	Resize X axis.
Change Sampling Rate	Change the sampling rate for the trends.
Zooming	Zoom in or out.
Scaling	Change the trend scales.
Stack Charts	Create new trend window with multiple stacked charts, one for each line trace in the active chart.
Reset	Reset trend window.
Statistics	Option to show lines for mean +/- n standard deviations.
Notes	Record notes at specific times in the data trends to save with the dataset.
Properties	Modify plotting properties such as titles, colors, borders, etc.
Print	Print chart.

Table 13 - Dataset Trends Right Click Menu Items

ABB. All rights reserved

The default is to provide one chart to trend the data. However, multiple charts can be displayed in the Trend window by specifying the desired number of charts in the corresponding selection box in the upper left side of the Trend window.
The user may then change the tags to display in each chart by selecting *Setup Trends* from the right click menu. This launches the *Assign Trend Taglists* window as shown in the following figure.

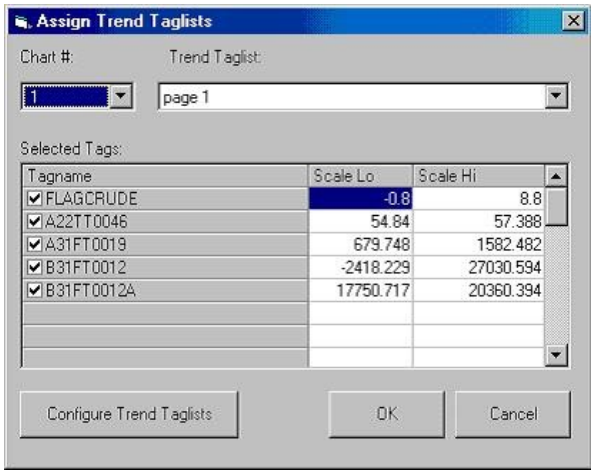


Figure 201 - Assign Trend Taglist

To display a particular trend taglist (group of tags) on a particular chart, select the desired chart number and then select the desired trend taglist and click **OK**. This step can be repeated for each chart.
The *Assign Trend Tagslist* window also includes a **Configure Trend Taglists** button that allows the user to edit existing trend taglists and to add new trend taglists. This button launches the *Configure Trend Taglists* window as shown in the following figure.

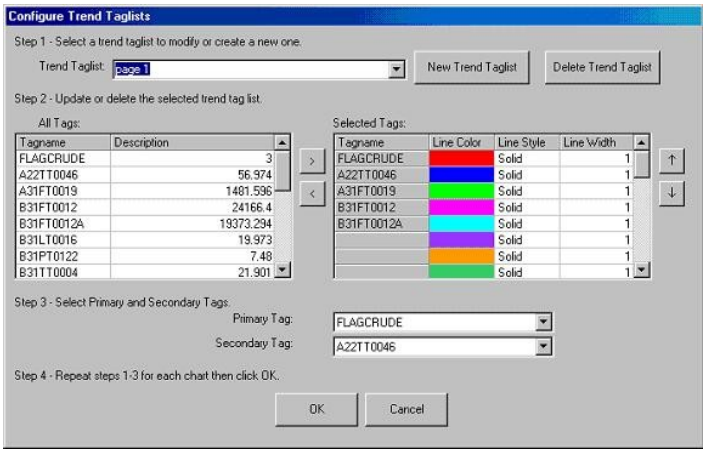


Figure 202 - Configure Trend Taglist

ABB. All rights reserved

The user should select a trend taglist to modify or click **New Trend Taglist** to create a new one.
The selected trend taglist on the right displays the tags in the selected trend taglist.
The user may delete a tag from this list by selecting it and clicking <. The user may add tags to the ***Selected Tags*** list by selecting the desired tag in the ***All Tags*** list on the left and clicking <.
Click **OK** to accept the changes.

ABB. All rights reserved

10. Appendix B – Transfer Functions

The modeling and tuning sections of this manual make references to Laplace transforms and transfer functions. Users familiar with these concepts do not need to read this section. For users not familiar with Laplace transforms and transfer functions, a very simple introduction is presented here to make the concepts used in LPM easier to understand.

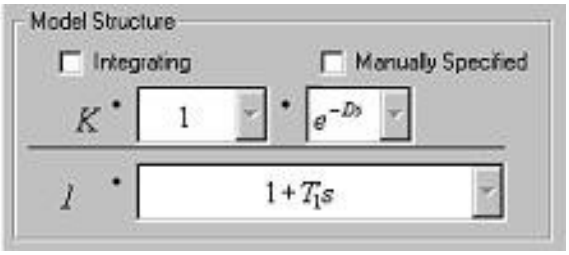


Figure 203 - Model Structure

Users not familiar with Laplace transforms who do not want to learn the details could skip this section and select first order models with deadtime for the model identification (like the selection shown in *Figure 203 - Model Structure*) and use the lambda tuning rule, if not dealing with level controllers or integrating loops.

To develop process models and to analyze controller tuning, control engineers create equations for process variables that are functions of time.

Consider as a process example a tank where the inlet and outlet flowrates, F1 and F2 respectively, affect the level, L. The equation describing the level is given by:

$$\frac{dL(t)}{dt} = k \bullet (F_1(t) - F_2(t))$$

In this case, the constant k is a function of the tank cross sectional area and the units of the level and the flows. The term dL/dt is read “the rate that level changes with time” and is known as the differential term.

This equation is known as a differential equation and to find L as a function of time, a special branch of calculus is used to solve such equations. In process control applications, where we add the controller equations and include feedback, the differential equations quickly get complicated and difficult to solve.

Control engineers use Laplace transforms as a shortcut to solve differential equations and to understand the type of dynamic response a system will have. For purposes of this discussion, a Laplace transform is created from a differential equation by substituting the Laplace operator “s” for the differential operator d/dt. Using Laplace transforms allows the engineer to combine a system a differential equations using standard algebra. There are special tables that help the engineer to convert a Laplace transform into a time equation without the d/dt term.

In the example above, replacing d/dt with s and solving for L,

$$L(s) = \frac{k \bullet (F_1(s) - F_2(s))}{s}$$

ABB. All rights reserved

The function k/s is called a ***transfer function***, since it denotes how the Laplace transforms for the inputs are transferred to that of the output. The 1/s form is typical of an integrator. If there is a step change in F1, then the response of L is a ramp. Control engineers know the shape of a step response for a given type of transfer function. Here are a few examples.

First order lag, T_1 = 60 seconds

$$\frac{1}{T_1 s + 1}$$

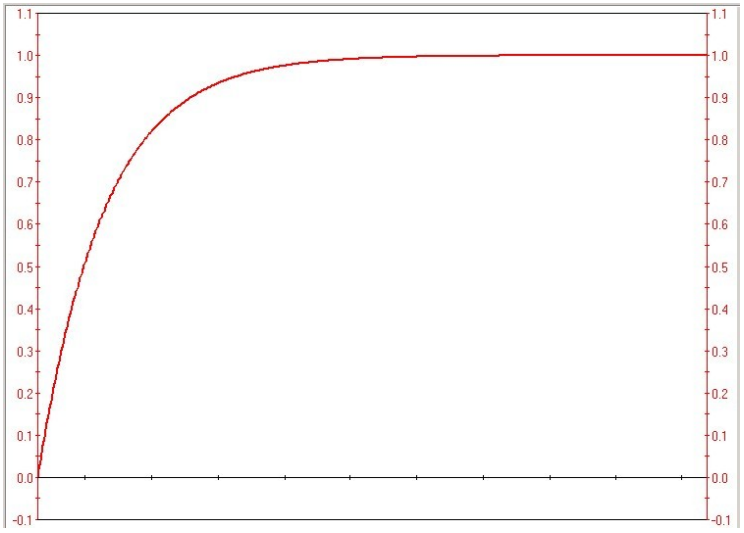


Figure 204 - First Order Lag

Second order lag, T1= T2 = 60

$$\frac{1}{(T_1 s + 1) \bullet (T_2 s + 1)}$$

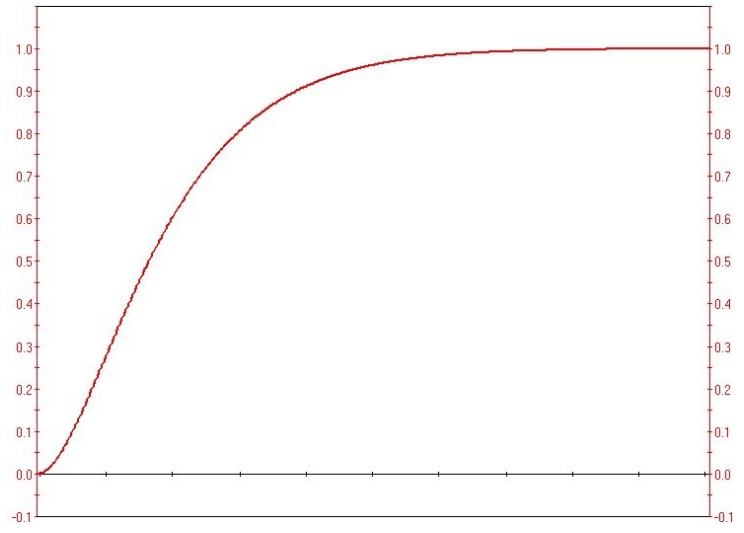


Figure 205 - Second Order Lag

Third order lag, T1 = T2 = T3 = 60

ABB. All rights reserved

$$\frac{1}{(T_1 s+1)\bullet(T_2 s+1)\bullet(T_3 s+1)}$$

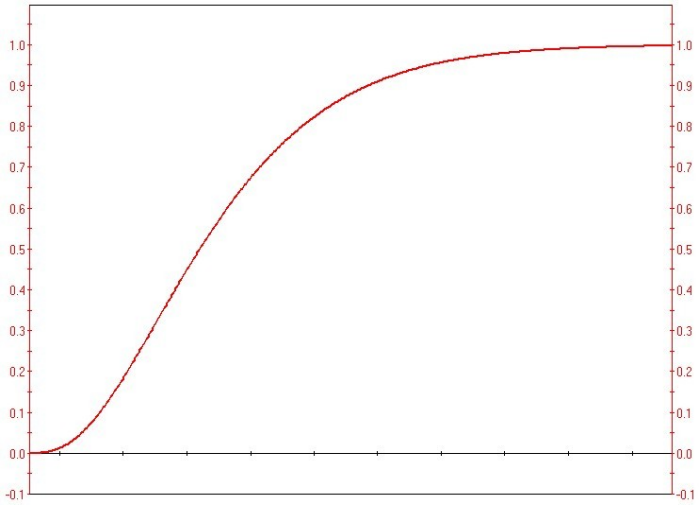


Figure 206 - Third Order Lag

Lead Lag T1=T2 = 60, Tz = 180. Notice, that since Tz is larger than both T1 and T2 and overshoot results without sinusoidal oscillations.

$$\frac{(T_z s+1)}{(T_1 s+1)\bullet(T_2 s+1)}$$

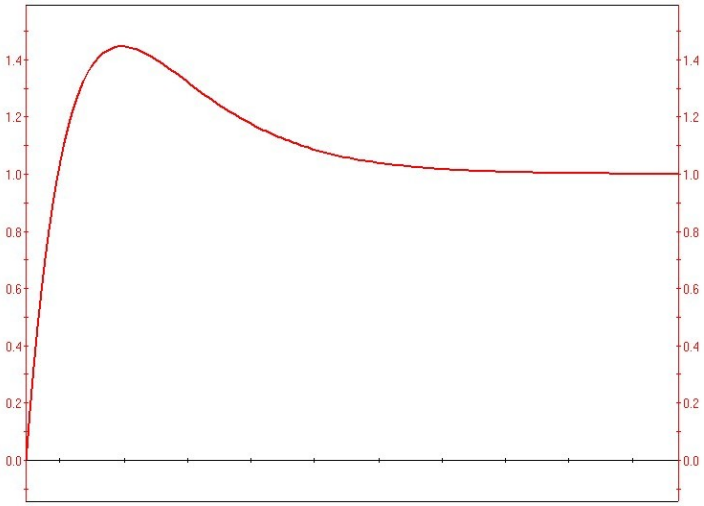


Figure 207 - Lead Lag

Because these functions specify the mapping between input signals and output signals, the rational groupings of polynomials in s are called “transfer” functions.

When modeling, the user should pick a Laplace transform structure that matches the expected behavior of the plant. If the plant responds similar to a first order lag, a third order lag should not be selected.

ABB. All rights reserved

There are a couple of important properties of Laplace transforms that control engineers use. In the block diagram below, $y(s) = G_2(s)*u(s)$ and $u(s) = G_1(s)*x(s)$. The rules of Laplace transform algebra allows the block diagrams to be simplified, so $y(s) = G_2(s)* G_1(s)*x(s)$.

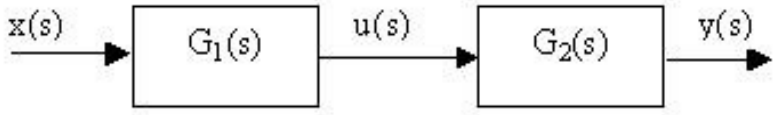


Figure 208 – Laplace Transform Representation

If we know the simple relationships from x to u and from u to y, then we know the more complex function-mapping x to y.
For analyzing feedback control, the block diagram structure includes a feedback loop. The Laplace transfer function from x to y can be found in two ways. One way is to use the algebra defined for block diagrams, called signal flow graphs. Another way is to define a variable at the output of the summing node, call it e(s). Find $y(s) = G_2(s)* G_1(s)*e(s)$ and $e(s) = y(s) – x(s)$.

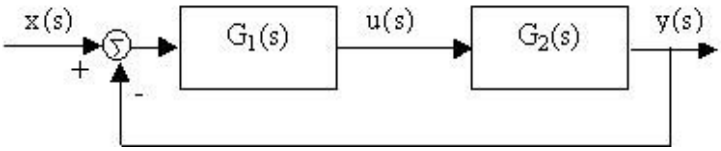


Figure 209 – Laplace Transfer Function for Analyzing Feedback Control

Substituting for e and solving for y, we get:

$$y(s) = \frac{G_1(s) \bullet G_2(s)}{1 + G_1(s) \bullet G_2(s)} x(s)$$

If G_1 is the transfer function for a PID controller and G_2 is the process model, then this equation represents the response of the closed loop system to a setpoint change.
There are three more special forms of Laplace transforms that are important for models that are encountered fairly often in process control applications. The first is deadtime. In terms of time, if a function F(t) is delayed by D seconds, the function becomes F(t-D). The Laplace transform for a time delay is the exponential term.

First order lag with deadtime. $T_1 = 60$, $D = 120$ seconds.

$$y(s) = \frac{G_1(s) \bullet G_2(s)}{1 - G_1(s) \bullet G_2(s)} x(s)$$

ABB. All rights reserved

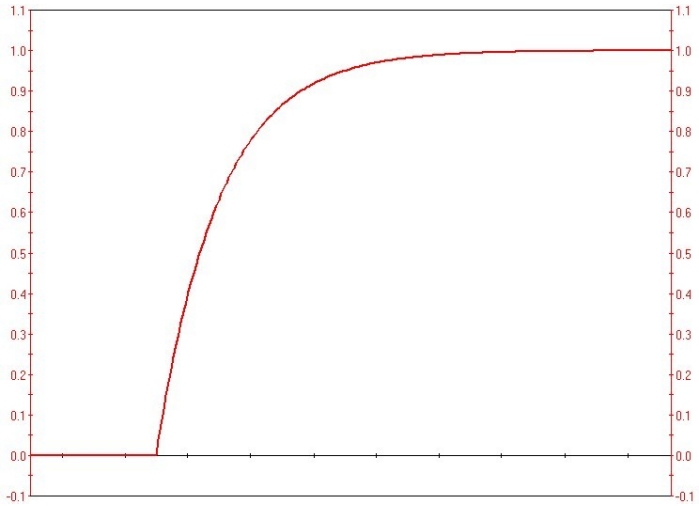


Figure 210 - First Order Lag with Deadtime

The second form of interest is called an inverse response. This type of function starts moving in one direction, but reverses and settles in the opposite direction of the initial response. The form is similar to the lead lag form seen earlier, but this time the time constant in the numerator is negative.

Inverse response lead lag. T1=T2 = 60, Tz = -180. The negative sign of Tz results in an initial response in the negative direction.

$$\frac{(T_z s + 1)}{(T_1 s + 1) \bullet (T_2 s + 1)}$$

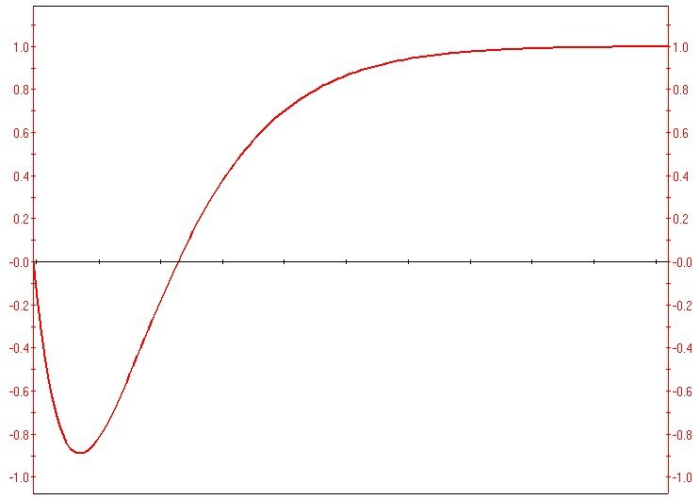


Figure 211 - Inverse Response lead lag

The final special form is the quadratic or second order system. The equation for this transfer function is:

ABB. All rights reserved

$$\frac{\omega^2}{(s^2 + 2\zeta\omega s + \omega^2)}$$

The response of this function depends on the values of ζ and ω. The figures below show the effect. The overshoot is controlled by ζ and ω sets the overall transient speed of response. For ζ < 1, the model is underdamped and includes oscillations.

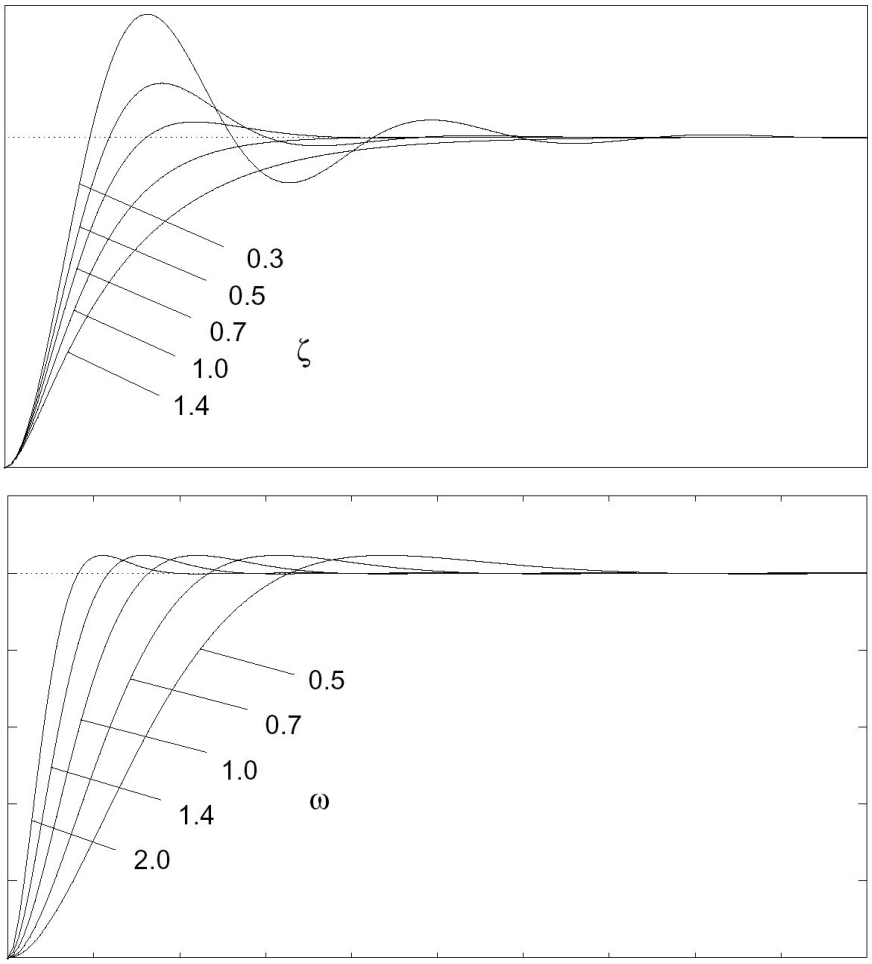


Figure 212 - Effect of ζ and ω values

ABB. All rights reserved

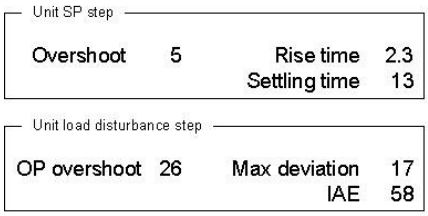
11. Appendix C – Technical Details

11.1 *Tuning Evaluation Parameters*

The following includes a list of tuning evaluation parameters together with a brief description.

11.1.1 Time response quantification

Example views:



11.1.1.1 Overshoot

The overshoot indicates the percentage for which the peak of the setpoint step response exceeds the final value

A high overshoot is an indication that the closed loop may not be robust enough.

11.1.1.2 Rise time

The rise time, t_r, is the time in seconds for the setpoint step response to change from 10% of 90% of its final value

The rise time is one of the indicators for the speed of the closed loop response.

11.1.1.3 Settling time

The settling time, t_s, is the time in seconds for the setpoint step response to settle to within ±5% of its final value

The shortest settling time, and also the fastest practical closed loop, is obtained when the tuning has been made as fast as it can without producing too much overshoot or becoming too oscillatory.

11.1.1.4 OP overshoot

The OP overshoot is how many percents the peak of the controller output response to a load step goes beyond the final value

For a positive final value of the controller output, the OP overshoot tells how many percents the peak exceeds the final value. For a negative final value it says how many percents below it goes.

ABB. All rights reserved

The value presented is normalized for a unit size load step, even if the response presented in the graph is shown for a different size.

11.1.1.5 Maximum deviation

Maximum deviation is the peak deviation (positive or negative) from the setpoint that the process value experiences during the load step response

The value presented is normalized for a unit size load step, even if the response presented in the graph is shown for a different size.

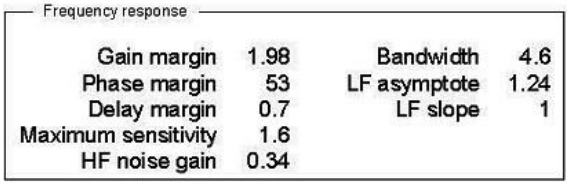
11.1.1.6 IAE

IAE, Integrated Absolute Error, is the total area between the setpoint value and the actual process value in the load step response plot

The absolute value of the deviation is integrated from the start of the upset until fully settled. Thus, areas of both positive and negative deviation contribute.
The value presented is normalized for a unit size load step, even if the response presented in the graph is shown for a different size.

11.1.2 Frequency response quantification

Example view:



11.1.2.1 Gain margin

Gain margin, A_m, is a factor telling how much the process gain would need to increase before the closed loop becomes unstable

If everything else is unchanged, but the process gain is multiplied by the gain margin, the closed loop is brought exactly to the stability limit.

11.1.2.2 Phase margin

Phase margin, φ_m, is how many degrees extra phase lag would need to be added to the process before the closed loop becomes unstable

If everything else is unchanged, but the phase lag of the process is increased by the phase margin, the closed loop is brought exactly to the stability limit.

11.1.2.3 Delay margin

Delay margin is how many seconds extra deadtime would need to be added to the process before the closed loop becomes unstable

ABB. All rights reserved

If everything else is unchanged, but the deadtime in the process is increased by the delay margin, the closed loop is brought exactly to the stability limit.

11.1.2.4 Maximum sensitivity

Maximum sensitivity, M_s, is the peak magnitude of the sensitivity function. High values indicate a non-robust closed loop.

M_s is always greater than 1, and should normally be less than 2. One interpretation of it is that the minimum distance to the critical point -1 in the Nyquist diagram is $1/M_s$.

11.1.2.5 HF noise gain

HF noise gain is the gain from measurement noise to controller output at high frequencies. High values lead to high actuator wear, and less robustness towards model inaccuracies at high frequencies.

The closed loop gain from measurement noise to controller output at the Nyquist frequency is what is presented as HF noise gain. The Nyquist frequency (π/T_s) is the largest frequency to consider in sampled control systems. All higher frequencies ought to be avoided by applying a suitable anti-alias filter to the process value. Beside the argument that actuators wear out more quickly with a large HF noise gain, models are often less accurate at high frequencies and therefore large controller gains at high frequencies should be avoided for robustness reasons. If the controller gain is large at high frequencies, also the presented HF noise gain is large, and vice versa.

11.1.2.6 Bandwidth

Disturbance rejection bandwidth, ω_B, in rad/s. Up to this frequency all disturbances are rejected by 3dB or more.

The sensitivity function has low amplitude values for low frequencies and the value 1 for high frequencies. For all frequencies below the disturbance rejection bandwidth, the amplitude of the sensitivity function is below -3dB ($1/\sqrt{2}$). This means that disturbances at those frequencies are rejected with at least the factor 3dB compared to the open loop situation.

11.1.2.7 LF asymptote

The value shown is the 'Error constant'. The higher the value, the better the low frequency disturbance rejection. Together with the slope, this value describes the low frequency asymptote of the sensitivity function.

The low frequency asymptote is described by ω^n/c_n, where ω is the angular frequency (rad/s), c_n is the error constant, and n is the slope. At frequency $\omega=1$, the sensitivity low frequency asymptote has the value $1/c_n$. For specific kinds of disturbances (step and ramp for example) one can calculate the stationary error using the error constant. If the slope is 0, a step disturbance of height h will give a stationary error h/c_0. If the slope is 1, a ramp disturbance with slope r will give a stationary error r/c_1.

ABB. All rights reserved

11.1.2.8 LF slope

The slope of the low frequency asymptote of the sensitivity function. This asymptote tells how well disturbances at low frequencies are rejected.

The slope of the low frequency asymptote of the sensitivity function tells how fast the disturbance rejection is improved when going from mid frequencies to low frequencies. It also tells whether for some kinds of disturbances, the stationary error vanishes or not. The stationary error resulting from a constant disturbance is zero if the slope is 1 or higher, but it is non-zero if the slope is 0. If the slope is 2 or higher, the stationary error resulting from a ramp disturbance is zero.

ABB. All rights reserved

12. Appendix D - Description of the Performance Indices

This chapter contains a brief description of all indices used for loop auditing in LPM. This chapter is helpful for the user, when interpreting the index values for loop assessment. The chapter contains description for each of the indices including limits, nominal values and hints how to interpret the index values.
Some of the indices used in LPM are rather used for documentation than for performance evaluation (e.g. loop category). For convenience some information was stored as indices.
Furthermore, a summary of all indices figures is given in a table at the end of the chapter.

Index number	Index name	Index number	Index name
1	Mean Loop Error	26	Oscillation amplitude regularity Index
2	Standard Deviation Loop Error (normalized)	27	Shut-off process value Index
3	Standard Deviation Loop Error (Eng. Unit)	28	Stiction Index 1
4	Mean Process Variable	29	Stiction Index 2
5	Standard Deviation Controller Output	30	Valve travel per hour
6	Skewness Control Error	31	Valve reversal per hour
7	Kurtosis Control Error	32	Valve sizing Index
8	Data compression Index	33	Non-linearity 1st Index
9	Data Quantization Index	34	Non-linearity 2nd Index
10	Outlier Index	35	Non-Gaussianity Index
11	Noise Level Index	36	Oscillation Symmetry Index
12	Ratio control error vs. controller output Index	37	Estimated process time-delay
13	Automatic Mode Index	38	Master Flag
14	Saturation Index	39	Setpoint step flag
15	Cascade Mode Index	40	Model fit flag
16	Auto-correlation ratio	41	Static Gain Index
17	Setpoint Crossing Index	42	Prediction horizon
18	Cascade Tracking Index	43	Loop Category
19	Cascade Travel Ratio Index	44	Area Tag
20	Harris Performance Index	45	User Defined Indices 1
21	Oscillation Index on Control error	46	User Defined Indices 2
22	Oscillation Index on Setpoint	47	User Defined Indices 3
23	Oscillation Period	48	Automatic Mode Index (Continuous)
24	Oscillation amplitude	49	Mean Absolute Loop Error (Continuous)
25	Oscillation severity Index	50	Saturated Percent (Continuous)

Table 14 – List of Indices

ABB. All rights reserved

12.1 *Mean Loop Error*

General data
LPM Name: mean_CE
Category: Basic Statistic

Description
The control error mean is an important measure of control performance. It has to be close to zero if no permanent off-set is present. In the index calculation the normalized control error (between 0 and 100 %) is used in order to make the quantities more comparable for different control loops.

Purpose
Assess the off-set of the (normalized) control error.

Interpretation
The value of the control error mean should be around zero. Any significant deviation indicates an off-set problem which could be e.g. due to missing integral control action or a sensor off-set. The quantification of what 'significant' means has to be assessed individually for each control loop.

Unit & range
Engineering unit, [-max, +max].

Nominal
The mean error should be as small as possible centered around zero.

Pre-requisite
None

Background
None

Reference
None

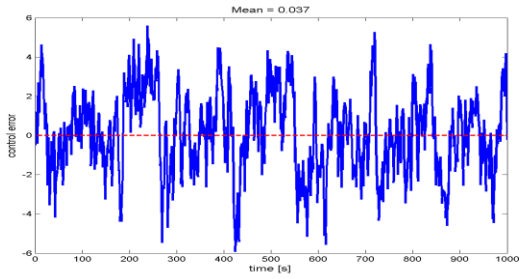
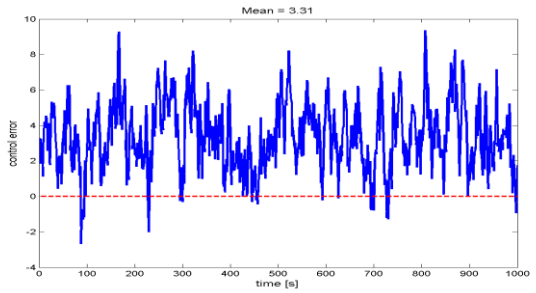
Good Situation	Bad Situation
	
Index = 0.007 t/h	Index = 3.31 t/h

ABB. All rights reserved

12.2 *Standard Deviation Loop Error (normalized)*

General data
LPM Name: stdev_CE_norm
Category: Basic Statistic

Description
The variation in the control error is the main target of automatic control. The loop error is aimed to be kept close to zero as good as possible.

Purpose
The standard deviation of the control error quantifies the average deviation from target.

Interpretation
The standard deviation of the control error is an absolute value. It is normalized to the loop range and hence enables comparison between different loops.

Unit & range
The loop error is in % of the loop range, [0, 100].

Nominal
If the loop ranges are reasonably chosen compared to the operating range, nominal values should be as small as possible, lower than 1-5%.

Pre-requisite
None

Background
Basic statistics

Reference
Standard textbooks on basic statistics.

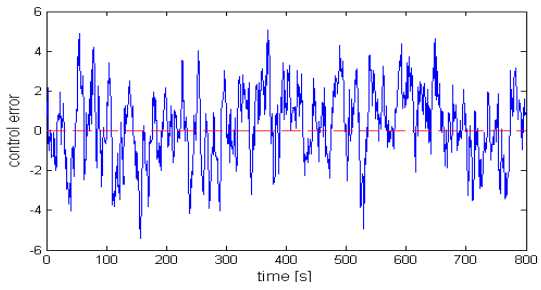
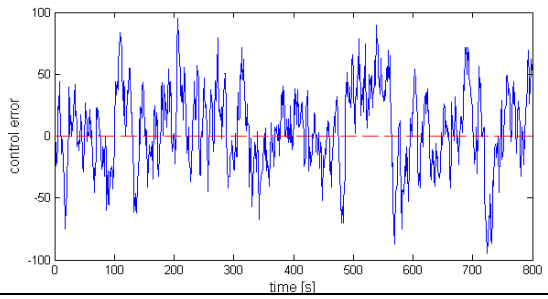
Good Situation	Bad Situation
	
Index = 0.74%	Index = 5.13%

ABB. All rights reserved

12.3 *Standard Deviation Loop Error (Eng. Unit)*

General data
LPM Name: stdev_CE_EngU
Category: Basic Statistic

Description
The same index as # 2 but in absolute engineering units.

Purpose
See above

Interpretation
See above

Unit & range
Engineering unit, [0, max].

Nominal
Nominal value should be small.

Pre-requisite
None

Background
See above

Reference
See above

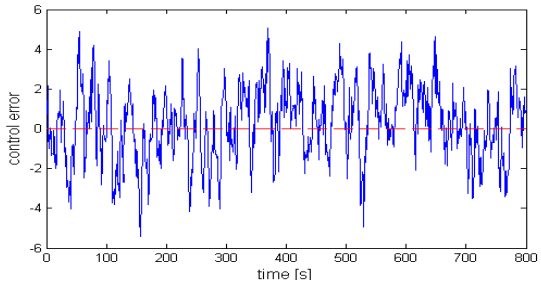
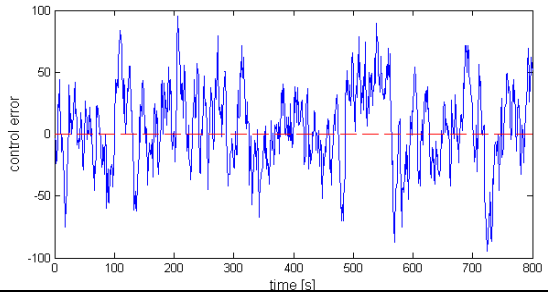
Good Situation	Bad Situation
	
Index = 2.03 t/h	Index = 32.5 t/h

ABB. All rights reserved

12.4 *Mean Process Variable*

General data LPM Name: mean_PV Category: Basic Statistic
Description The mean process value documents the actual operating point that was used during data collection. This index is no performance index but important for analysis.
Purpose The mean process value documents the current operating point.
Interpretation There is no performance related interpretation. The mean value may be used to analyze if certain control problems are related to certain operating points.
Unit & range The mean process value can take any value from the operating range, [min, max].
Nominal Not applicable
Pre-requisite None
Background None
Reference None

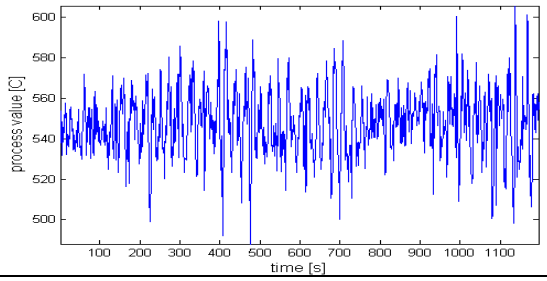
Example	
	**Not applicable**
Index = 547 °C	

ABB. All rights reserved

12.5 *Standard Deviation Controller Output*

General data LPM Name: stdev_OP Category: Basic Statistic
Description The standard deviation of the controller output is a measure for the control effort required to keep the loop under automatic control.
Purpose The standard deviation and its change reveals control effort.
Interpretation Standard deviation of the control error may quantify the actuator effort.
Unit & range The standard deviation has the same unit as the controlled variable itself. Mostly it is [0%, 100%].
Nominal Not applicable, a very high value typically indicates a loop problem.
Pre-requisite None
Background None
Reference None

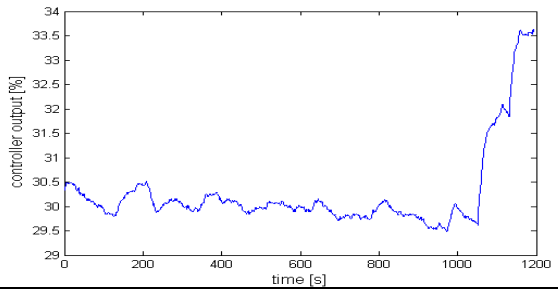
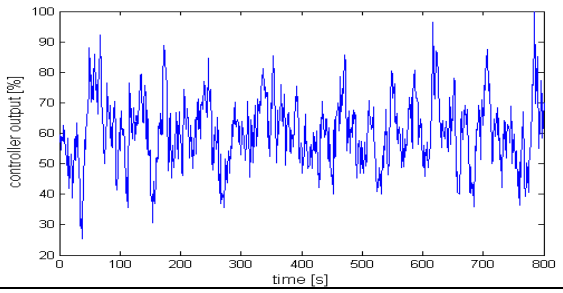
Good Situation	Bad Situation
	
Index = 0.89	Index = 13.6

ABB. All rights reserved

12.6 *Skewness Control Error*

General data
LPM Name: skew_CE_norm
Category: Basic Statistic

Description
The skewness is the third central moment divided by the cube of the standard deviation. Skewness is a measure of the asymmetry of the data around the sample mean. If skewness is negative, the data are spread out more to the left of the mean than to the right. If skewness is positive, the data are spread out more to the right. The skewness of the normal distribution (or any perfectly symmetric distribution) is zero.

$$s = \frac{E(x-\mu)^3}{\sigma^3}$$

Purpose
Assess the symmetry of the control error.

Interpretation
The skewness is an important indicator of control problems, especially for nonlinear phenomena.

Unit & range
Dimensionless, [-max, +max].

Nominal
Non-skew data has a skewness of zero. Significant if outside [-1, +1].

Pre-requisite
None

Background
Basic statistic theory.

Reference
Any textbook on basic statistics.

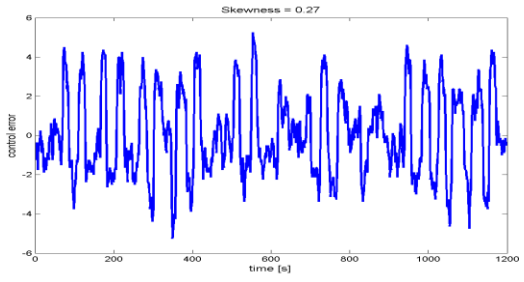
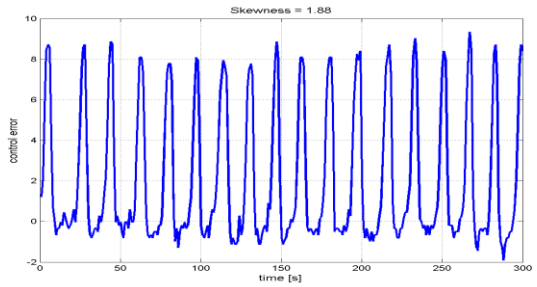
Good Situation	Bad Situation
	
Index = 0.27	Index = 1.88

ABB. All rights reserved

12.7 *Kurtosis Control Error*

General data
LPM Name: kurt_CE_norm
Category: Basic Statistic

Description
The kurtosis is the fourth central moment divided by fourth power of the standard deviation. Kurtosis is a measure of how outlier-prone a distribution is.

$$k = \frac{E(x-\mu)^4}{\sigma^4} - 3$$

Purpose
The kurtosis is a measure for the flatness of the data.

Interpretation
It quantifies if the data has larger or smaller tails than a Gaussian distributed signal. The kurtosis for Gaussian data is theoretically 0. Any significant deviation from this value indicates a flatter or thinner distribution of data. Therefore, distributions that are more outlier-prone than the normal distribution have kurtosis greater than 0; distributions that are less outlier-prone have kurtosis less than 0.

Unit & range
The kurtosis is dimensionless. The value ranges from [-max, +max].

Nominal
Normally distributed data has a kurtosis of 0. Good values are within [-1.5, 1.5].

Pre-requisite
None

Background
Basic statistic theory.

Reference
Any textbook on basic statistics.

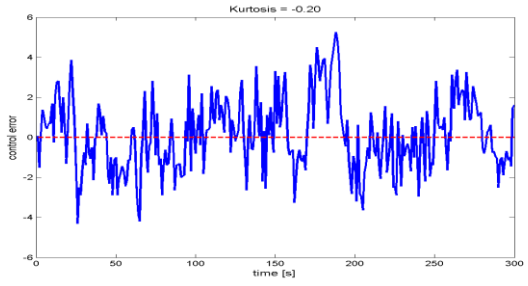
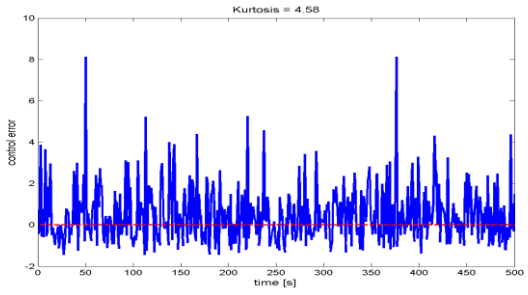
Good Situation	Bad Situation
	
Index = -0.20	Index = 4.58

ABB. All rights reserved

12.8 *Data compression Index*

General data
LPM Name: compression
Category: Data Validity

Description
If auditing data is obtained from a data historian, compression might have been applied to the data. This index quantifies the amount of compression within the data. Auditing may be unreliable if the data had been compressed. This is true even for simple statistics as the data mean.

Purpose
Quantify the amount of compression in the data set in order to assess if the analysis of the data is reliable.

Interpretation
If the index is greater than the threshold, the analysis results may not be reliable and should not be trusted.

Unit & range
The index is dimensionless and the range is [0...Inf].

Nominal
Nominal values are below 3.

Pre-requisite
None

Background
The algorithm looks for linear segments in the data as typically found in compressed data. It will not work for data that has been compressed using e.g. splines.

Reference
Nina F. Thornhill a,*, M.A.A. Shoukat Choudhury b, Sirish L. Shah (2004), The impact of compression on data-driven process analyses. Journal of Process Control.

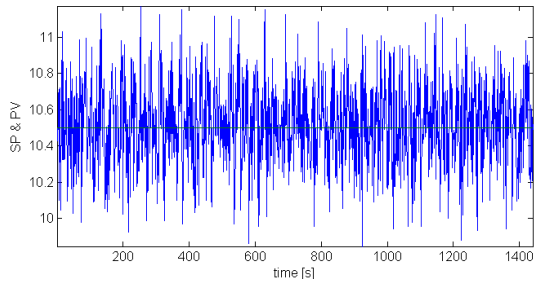
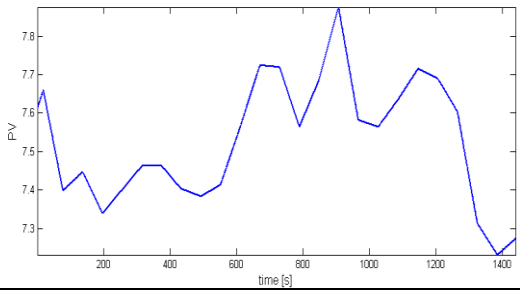
Good Situation	Bad Situation
	
Index = 2	Index = 41.7

ABB. All rights reserved

12.9 *Data Quantization Index*

General data
LPM Name: quantization
Category: Data Validity

Description
This index quantifies the amount of quantization in the data. Quantization due to e.g. low resolution in AD-converters imposes a static nonlinearity on the control loop. If the quantization is too strong, control performance may be reduced. The index is defined as the percentage of the actual quantization level w.r.t the control error standard deviation. Also, the relationship with the measurement range is taken into account.

Purpose
Detect performance problems caused by signal quantization.

Interpretation
If the quantization level is lower than a 10 bit resolution (100/1024), then the index value is 0. Otherwise, the percentage describes the ratio above. A value below 40% is usually acceptable.

Unit & range
[0...100%]

Nominal
Below 40%.

Pre-requisite
None

Background
Heuristic index

Reference
None

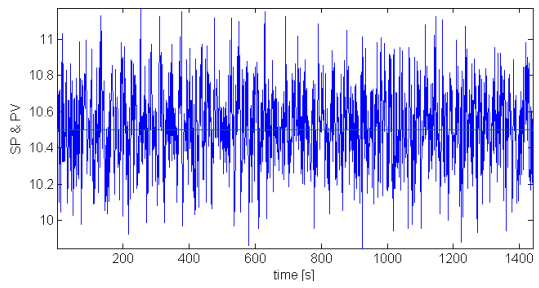
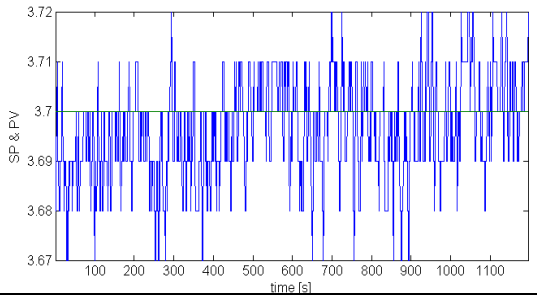
Good Situation	Bad Situation
	
Index = 0%	Index = 80.1%

ABB. All rights reserved

12.10 *Outlier Index*

General data
LPM Name: outlier
Category: Data Validity

Description
A simple outlier detection based on a sliding window that detects single outliers that exceed the standard deviation significantly. The significance is detected using a median filter. Only single off-values are considered outliers.
Identified outliers are automatically removed from the data and replace by linear regression.

Purpose
Detect and remove outliers automatically.

Interpretation
The index reports the amount of single outliers w.r.t to the total number of samples in the data batch.

Unit & range
[0...1000 ‰]

Nominal
0 ‰, i.e. no outlier.

Pre-requisite
None

Background
Very simple yet robust method for outlier detection.

Reference
None

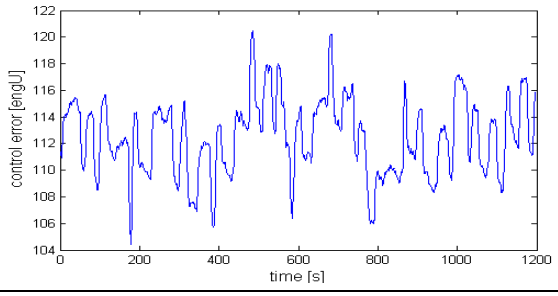
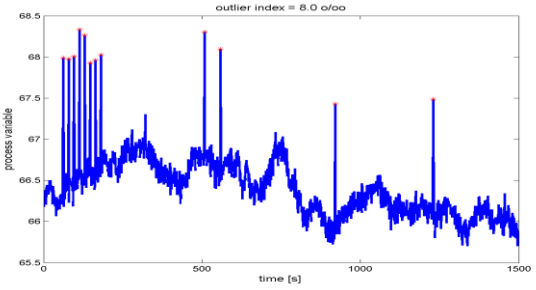
Good Situation	Bad Situation
	
Index = 0‰	Index = 8‰

ABB. All rights reserved

12.11 *Noise Level Index*

General data
LPM Name: noise_strength
Category: Loop Modes

Description
The noise index quantifies the amount of noise in the control error signal. The actual noise level is computed and compared to the control error standard deviation. The noise level estimation is based on wavelet decomposition.

Purpose
Quantify amount of noise in the control error.

Interpretation
The higher the value, the more dominant is the noise in the control error. If the value is low, other than stochastic variability is dominant in the signal (e.g. oscillation).

Unit & range
Percentage of noise in control error compared standard deviation [0, 100%].

Nominal
Ideally only stochastic variation should be presented, hence a high value of the index indicates good operation (Index>50%).

Pre-requisite
Signal non-constant.

Background
The noise level estimation is based on advanced wavlet signal decomposition.

Reference
ABB Patent DE10225344A1

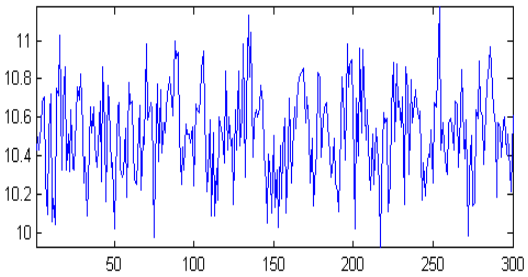
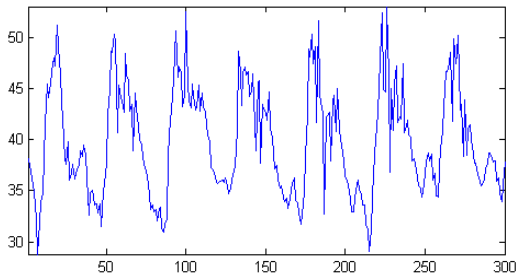
Good Situation	Bad Situation
	
Index = 100%	Index = 6.4%

ABB. All rights reserved

12.12 *Ratio control error vs. controller output Index*

General data
LPM Name: ratio_CE_CO
Category: Loop Modes

Description
The price for keeping the control error at zero is variability in the controller output. The index quantifies the standard deviations of the (normalized) control error and the controller output.

Purpose
Quantify where the variability is located in a control loop.

Interpretation
A large value indicates too little control effort resulting in large variation in the process value. A small value indicates heavy control effort with very efficient control. Values around one include normal operation but no guarantee for good control. Important is also the change in the index over time.

Unit & range
Dimensionless: [0...Inf]

Nominal
All values those are not extremely large or small.

Pre-requisite
None

Background
None.

Reference
None

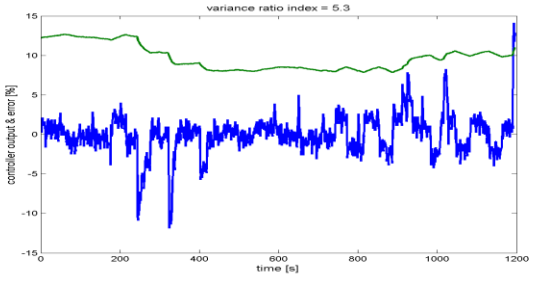
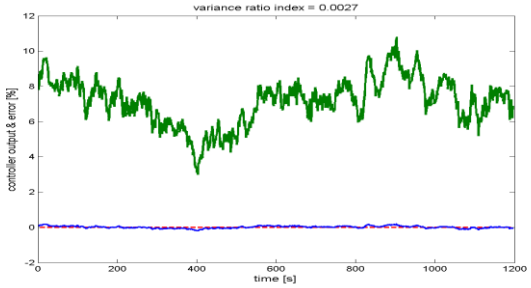
Good Situation	Bad Situation
	
Index = 5.3	Index = 0.0027

ABB. All rights reserved

12.13 *Automatic Mode Index*

General data LPM Name: mode_automatic Category: Loop Modes
Description Based on the controller output, this index quantifies the amount of time the control loop is in automatic mode. The index checks the variability of the controller output.
Purpose Track service factor of the control loop.
Interpretation Amount of samples in automatic per data batch.
Unit & range [0...100%] of samples in data batch.
Nominal 100%
Pre-requisite None
Background The interpretation of the MODE tag in different DCSs is sometimes not straightforward. This method uses the controller output to estimate the Auto Mode percent of time and is independent from the specific DCS type.
Reference None

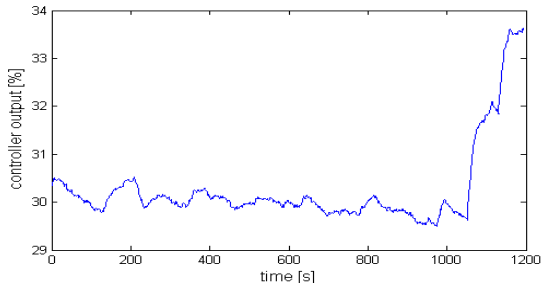
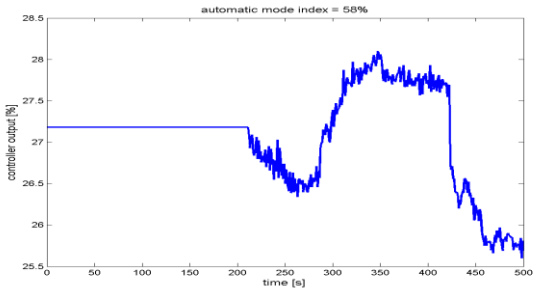
Good Situation	Bad Situation
	
Index = 100%	Index = 56%

ABB. All rights reserved

12.14 *Saturation Index*

General data LPM Name: mode_saturation Category: Loop Modes
Description The index quantifies times where the controller output stays significantly long below 3% or above 97% of the controller ouput range.
Purpose Quantify amount of time the loop is in saturation.
Interpretation Percentage of saturation in control loop.
Unit & range [0...100%] of samples in data batch.
Nominal 0%
Pre-requisite None
Background None.
Reference None

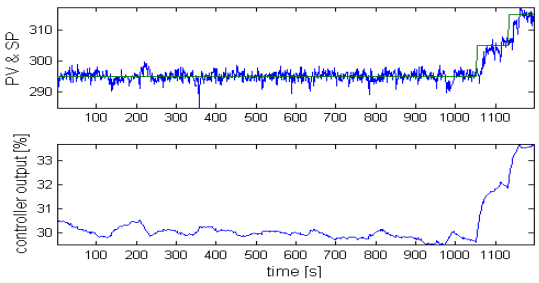
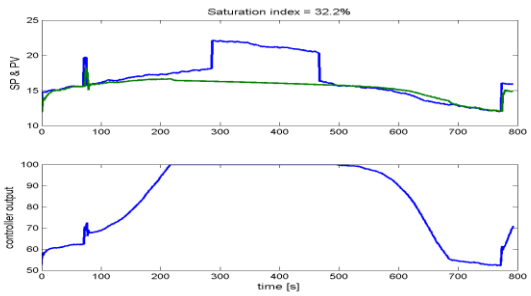
Good Situation	Bad Situation
	
Index = 0%	Index = 32.2%

ABB. All rights reserved

12.15 *Cascade Mode Index*

General data LPM Name: mode_cascade Category: Loop Modes
Description This index is not directly related to controller performance. It describes if the loop setpoint is changed continuously or stepwise (or not at all).
Purpose Determine if the loop is within a cascade (from the setpoint signal).
Interpretation The times of setpoint movement is quantified.
Unit & range [0...100%] of samples in data batch.
Nominal Not applicable
Pre-requisite None
Background None.
Reference None

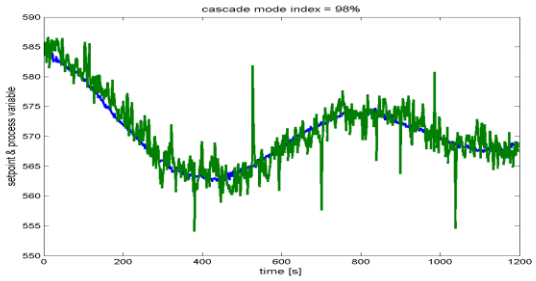
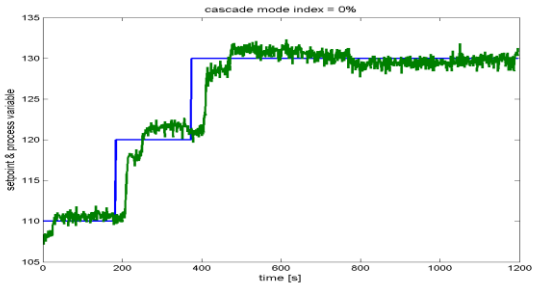
Cascade Control	Not Cascade Control
	
Index = 98%	Index = 0%

ABB. All rights reserved

12.16 *Auto-correlation ratio*

General data
LPM Name: ACF_ratio_index
Category: Performance

Description
The index describes how fast the auto-correlation (ACF) of the data decays over time. The value where the ACF decreased below 30% is compared with the given loop category individual horizon. The ACF index is defined as the ratio between the prediction horizon and the lag where the ACF falls below 30%.

Purpose
Quantify sluggishness of control loop responses.

Interpretation
The index is the ratio of the horizon and the actual ACF 30% value. Hence a value close to 0 means that the ACF decays much slower than the benchmark. A value close to 1 means that the ACF decays as fast as the benchmark or faster.

Unit & range
[0....1]

Nominal
Values close to 1.

Pre-requisite
It is necessary the Loop is working in Automatic Mode and the number of samples acquired is large enough (N>200).

Background
Sample auto-correlation function and Harris performance index.

Reference
None

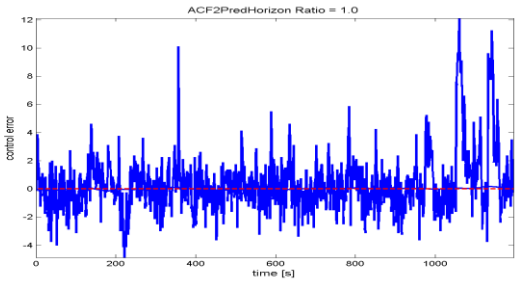
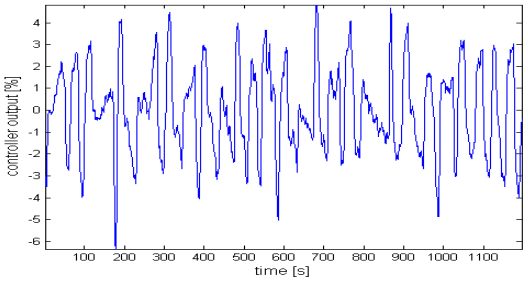
Good Situation	Bad Situation
	
Index = 1	Index = 0.1

ABB. All rights reserved

12.17 *Setpoint Crossing Index*

General data LPM Name: crossing_SP Category: Performance
Description Amount of control error zero crossings related to the total number of samples in the data batch.
Purpose Quantify the closeness of the control error to zero.
Interpretation A low value means long periods where the control error is either above or below zero. White noise has a theoretical value of 50%. A value >10% is considered to be good. The index is not used for diagnosis of level loops.
Unit & range [0...100%] of zero-crossings in data batch.
Nominal >10 %
Pre-requisite None
Background none
Reference Michael A. Paulonis, John W. Cox. "A practical approach for large-scale controller performance assessment, diagnosis, and improvement". In Journal of Process Control 13 (2003) 155–168.

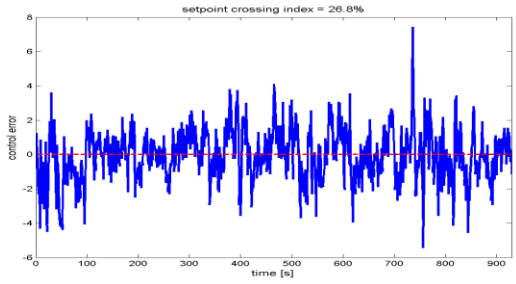
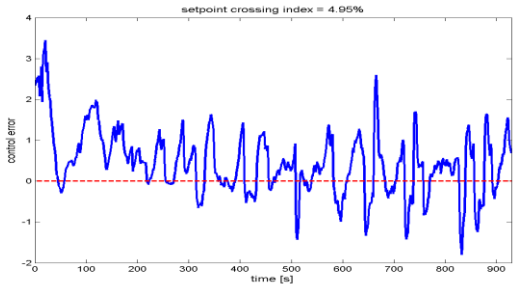
Good Situation	Bad Situation
	
Index = 26.8%	Index = 4.95%

ABB. All rights reserved

12.18 *Cascade Tracking Index*

General data
LPM Name: tracking_SP
Category: Performance

Description
A simple index that counts all samples that deviate more than 3% (of the loop range) from the desired setpoint value. The index is defined as the number of samples outside the specification divided by the total number of samples.

Purpose
Quantify setpoint tracking performance.

Interpretation
A high value means that the control error is often far away (more than 3% of loop range) from the target. A low value means that the control succeeds in holding the process variable close to the target.

Unit & range
[0...100%]

Nominal
<20%

Pre-requisite
Continuously moving setpoint.

Background
none

Reference
None

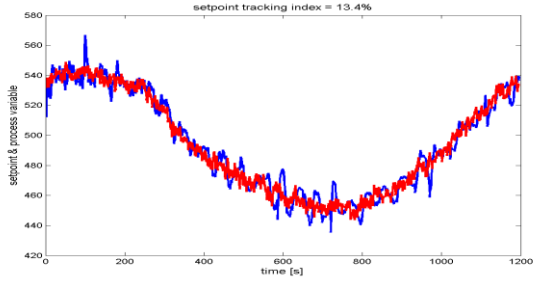
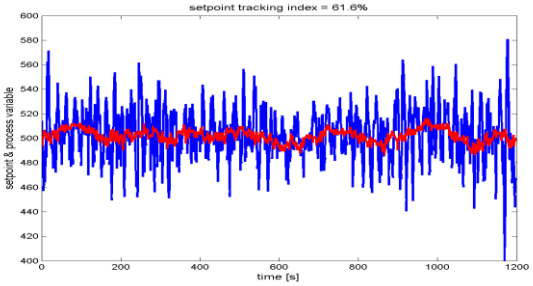
Good Situation	Bad Situation
	
Index = 13.4%	Index = 61.6%

ABB. All rights reserved

12.19 *Cascade Travel Ratio Index*

General data
LPM Name: track_ratio_SP_PV
Category: Performance

Description
This index relates the "travel" per hour for the setpoint with the one for the process value. For good control, both signals should "travel" equally much. Travel here refers to the integral of the signal after it has been cleaned from noise.

Purpose
Asset cascade setpoint tracking.

Interpretation
For equal tracking the index should be around 1.

Unit & range
[0...Inf] dimensionless.

Nominal
[0.5...1.5]

Pre-requisite
Loop have to be operating in Cascade Control (Casc_mode>30%)

Background
None

Reference
None

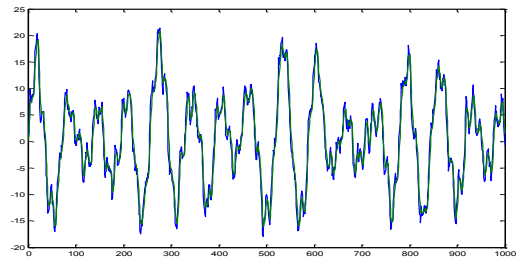
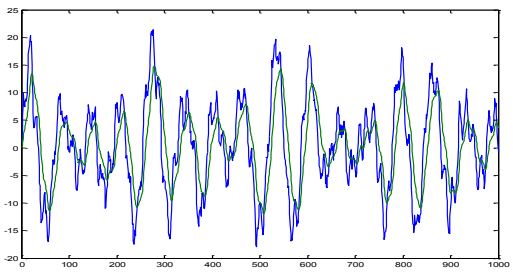
Good Situation	Bad Situation
	
Index = 1.39	Index = 2.43

ABB. All rights reserved

12.20 *Harris Performance Index*

General data
LPM Name: control_performance
Category: Performance

Description
Absolute measure of control performance with respect to attenuation of stochastic disturbances. For each control loop category, an individual benchmark is set.

Purpose
Assess controller performance based on operating data.

Interpretation
The closer the index is to 1, the closer the actual performance is to the benchmark. The inverse of the index indicates the factor by which the variance of the loop can be decreased compared to the benchmark, i.e. 0.33 would mean factor of 3.

Unit & range
Dimensionless, [0, 1]

Nominal
The range of good control is [0.5, 1].

Pre-requisite
Data must not be constant. Loop is working in Automatic Mode and the number of samples acquired is large enough (N>200).

Background
Basic theory on stochastic control theory.

Reference
T. Harris (1989), ‚Assessment of control loop performance‘. The Canadian Journal of Chemical Engineering 67, 856-861.

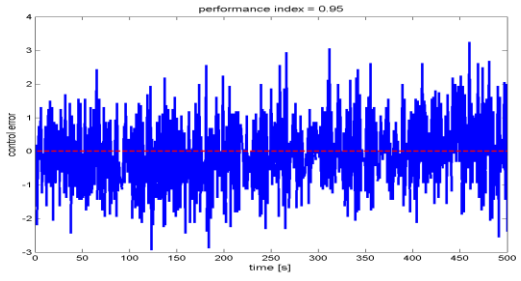
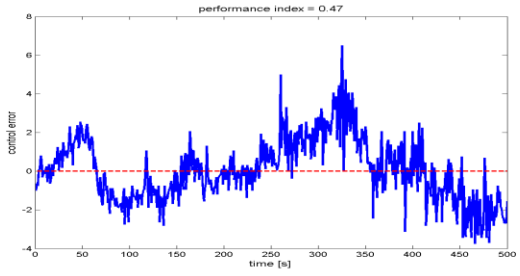
Good Situation	Bad Situation
	
Index = 0.96	Index = 0.47

ABB. All rights reserved

12.21 *Oscillation Index on Control error*

General data
LPM Name: oscillation_CE
Category: Oscillation

Description
Detection of oscillations in control loops is a central cornerstone in control loop monitoring. The oscillation detection based on the control error signal is the main pre-requisite for several diagnosis indices, which are evaluated succeeding the detection.

Purpose
Detect oscillatory control error behavior.

Interpretation
The oscillation index both detects and quantifies oscillatory signals. If the index is negative, not oscillation was found. If the index is positive, an oscillation was found. The closer to 1 the index is, the more regular the oscillations are.

Unit & range
The index is dimensionless and the range is [-1, 1].

Nominal
Nominal values (i.e. no oscillatory behavior) is an index smaller than 0. If the Index is larger than 0.5, there are oscillation and they are quite regular.

Pre-requisite
The loop has to be in automatic mode. The number of samples acquired has to be large enough (N>200).

Background
The index makes use of the zero-crossings of the auto-covariance function of the control error. The regularity of the zero-crossings is transformed to the index above.

Reference
N.F. Thornhill, T. Hägglund, Detection and diagnosis of oscillation in control loops, Control Eng. Prac. 5, (1997) 1343-1354.

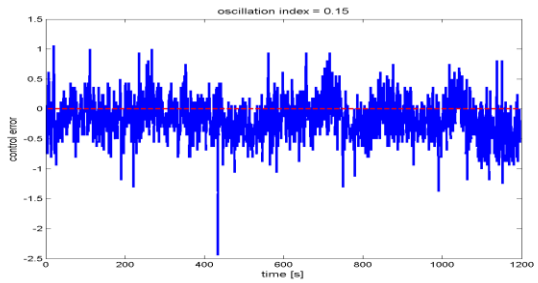
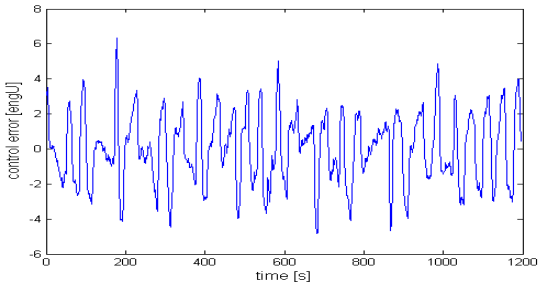
Good Situation	Bad Situation
	
Index = -0.44	Index = 0.81

ABB. All rights reserved

12.22 *Oscillation Index on Setpoint*

General data
LPM Name: oscillation_SP
Category: Oscillation

Description
The detection of setpoint oscillations is important for the detection of external excitation in a control loop. The setpoint is usually much less distorted and noisy than the control error. The setpoint can only oscillate if it is the output a master controller in a cascade loop.

Purpose
Detect (external) setpoint oscillations in a control loop.

Interpretation
The oscillation index both detects and quantifies oscillatory signals. If the index is negative, not oscillation was found. If the index is positive, an oscillation was found. The closer to 1 the index is, the more regular the index is.

Unit & range
The index is dimensionless and the range is [-1, 1].

Nominal
Nominal values (i.e. no oscillatory behavior) is an index smaller than 0. If the Index is larger than 0.5, there are oscillation and they are quite regular.

Pre-requisite
The loop has to be in automatic mode. The number of samples acquired has to be large enough (N>200).

Background
For the oscillation detection of setpoint signals, the regularity of zero-crossings and integrated absolute errors of half-periods (IAEs) has been used and is combined in a single index. The index is scaled such that the interpretation is the same as for the oscillation index of the control error.

Reference
N.F. Thornhill, T. Hägglund, Detection and diagnosis of oscillation in control loops, Control Eng. Prac. 5, (1997) 1343-1354.

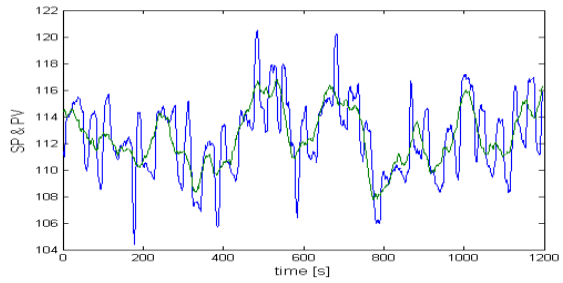
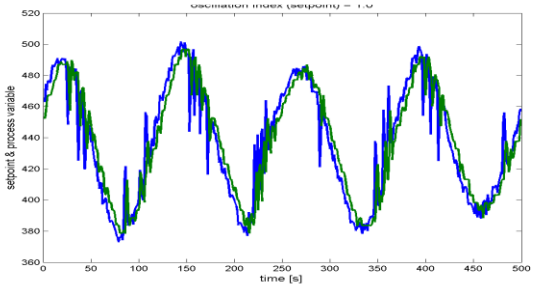
Good Situation	Bad Situation
	
Index = -0.24	Index = 0.93

ABB. All rights reserved

12.23 *Oscillation Period*

General data LPM Name: osc_period_time Category: Oscillation
Description When an oscillation has been detected, it is useful for more detailed analysis to have an estimate of the average oscillation period of that oscillation.
Purpose Store the estimated oscillation period of an oscillating control loop.
Interpretation Main oscillation period in seconds.
Unit & range [0...Inf] seconds
Nominal Not applicable
Pre-requisite Oscillation detected. Enough data (N>200).
Background See oscillation detection. The oscillation period is estimated from the signal auto-correlation function which is strongly de-noised and has the same oscillation period as the time-domain data.
Reference N.F. Thornhill, T. Hägglund, Detection and diagnosis of oscillation in control loops, Control Eng. Prac. 5, (1997) 1343-1354.

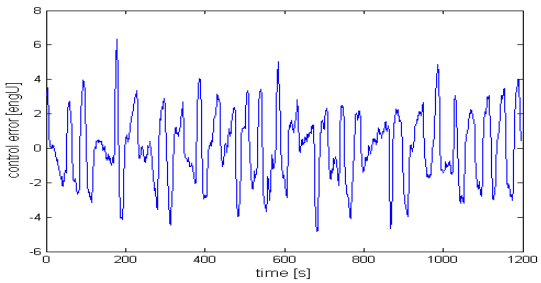
Example	
	
Index = 41.1 [s]	

ABB. All rights reserved

12.24 *Oscillation amplitude*

General data LPM Name: osc_amplitude Category: Oscillation
Description When an oscillation has been detected, it is useful for more detailed analysis to have an estimate of the average oscillation amplitude of that oscillation.
Purpose Store the estimated oscillation amplitude of an oscillating control loop.
Interpretation Average amplitude in percentage of loop range.
Unit & range [0...100%]
Nominal Not applicable
Pre-requisite Oscillation detected. Enough data (N>200).
Background The oscillation amplitude is estimated from the amplitude spectrum of the data.
Reference None

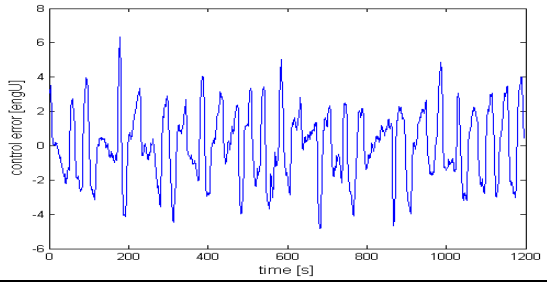
Example	
	
Index = 0.72%	

ABB. All rights reserved

12.25 *Oscillation severity Index*

General data
LPM Name: osc_severity
Category: Oscillation

Description
When an oscillation has been detected, it is of interest how much of the signal power is related to that single oscillation after noise effects have been removed. Harmonics of an oscillation have been taken into account.

Purpose
Assess the severity of a detected oscillation.

Interpretation
The index presents the percentage of variability of the main oscillation to the total signal power.

Unit & range
[0...100%]

Nominal
Not applicable

Pre-requisite
Oscillation has been detected. Enough data (N>200).

Background
Signal spectrum estimation.

Reference
Standard textbook on signal processing.

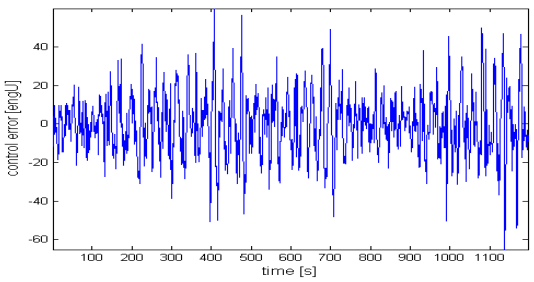
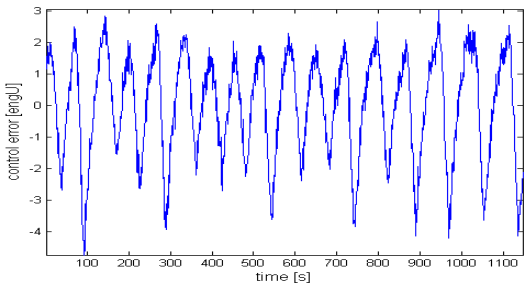
Good Situation	Bad Situation
	
Index = 46.6%	Index = 70.8%

ABB. All rights reserved

12.26 *Oscillation amplitude regularity Index*

General data
LPM Name: osc_regularity
Category: Oscillation

Description
An important characteristic of an oscillation is the regularity of its amplitude. This index quantifies to what extend the amplitude of a detected oscillation varies over the batch. The index describes the ratio between the standard deviation of the amplitudes of all half-periods and their mean. In order to make the index intuitive, the ratio is subtracted from 1 and given in %.

Purpose
Quantify the amplitude regularity for an oscillating loop.

Interpretation
The higher the index the more regular the oscillation amplitude is, i.e. the lower is the standard deviation of the amplitudes in each half-period. Hence an index of 100% means that the amplitude is the same in each period and 0% means that the variation is in the range of the amplitude itself.

Unit & range
The index is given in percent and is [0, 100]%.

Nominal
Since oscillation is always problematic, there is not , good value. An oscillation amplitude is considered non-regular if the index drops below 60%.

Pre-requisite
Oscillation has been detected. Enough data (N>200).

Background
Irregular oscillation amplitudes are often an indication for tuning problems.

Reference
None

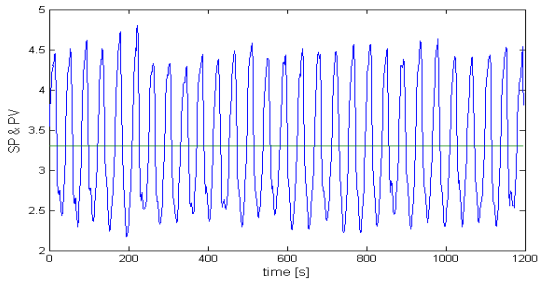
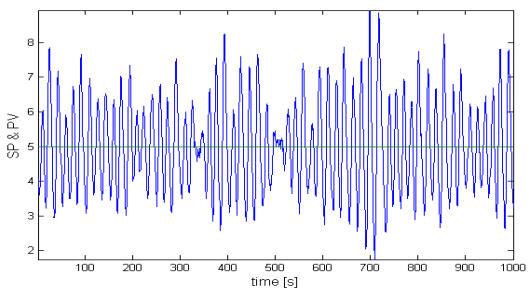
Good Situation	Bad Situation
	
Index = 90%	Index = 53%

ABB. All rights reserved

12.27 *Shut-off process value Index*

General data
LPM Name: shutoffvalue_PV
Category: Valve

Description
If the controller output is 0%, there should not be any measurable flow in a flow control loop. The valve is supposed to be closed in this case. A significant non-zero flow should not be present. Reasons can be either valve leakage or a wrong instrument calibration.

Purpose
Detect valve leakage or zero instrument errors.

Interpretation
The index describes the average value in the process variable while the controller output is zero.

Unit & range
The index gives the average process value [% of operating range] during CO=0%.

Nominal
Normally, the index should be 0%, which means that no flow can be measured if the controller output is zero.

Pre-requisite
Controller output is zero for some significant time.

Background
None

Reference
None

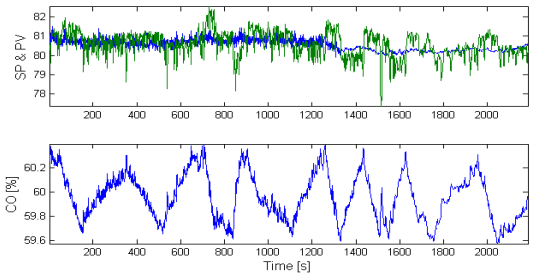
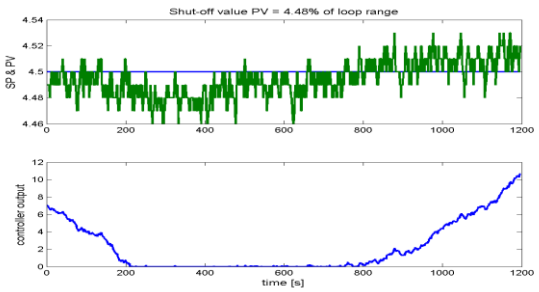
Good Situation	Bad Situation
	
Index = 0%	Index = 4.48%

ABB. All rights reserved

12.28 *Stiction Index 1 (Cross correlation Method)*

General data
LPM Name: flag_stiction1
Category: Valve

Description
When an oscillation has been detected, it is desired to check for stiction as one of the possible problems. Stiction can be detected (assumed the loop is oscillating, as indicated by the oscillation index) by considering the cross-correlation function between control error and controller output. This algorithm is only used for self-regulating loops and non-compressible fluids. The method may yield wrong results for integrating loops (level loops) or steam / gas flow loops. For these cases, the alternative method for stiction detection is used.
In cases where both methods apply, both are used for hypothesis generation.

Purpose
Detection of valve stiction.

Interpretation
If the algorithm finds indications for stiction, the index returns +1, if the algorithms find indications for either external (oscillating) disturbances or internal loop problems (most likely too tight a controller tuning), the index returns 0. If no safe decision can be made, the index returns the value –1.

Unit & range
[0,1], = -1 if no safety decision can be taken.

Nominal
No stiction = 0.

Pre-requisite
Loop has been detected as oscillating.

Background
Theory of stiction generation and control loops.

Reference
A.Horch. „A simple method for detection of stiction in process control loops.“ In Control Eng. Pract., 7(10), pp.1221-31, 1999.

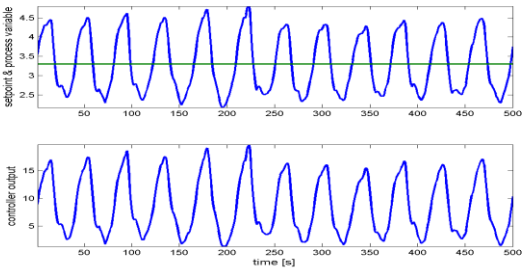
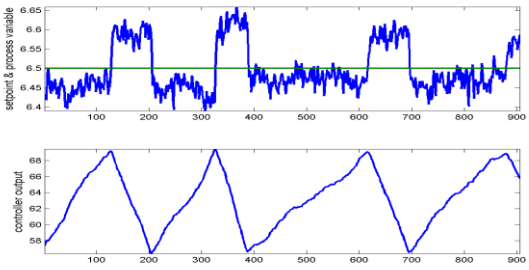
Good Situation	Bad Situation
	
Index = 0	Index = 1

ABB. All rights reserved

12.29 *Stiction Index 2 (Histogram method)*

General data
LPM Name: flag_stiction2
Category: Valve

Description
When an oscillation has been detected, it is desired to check for stiction as one of the possible problems. This method is based on a "Histogram Method". See the relevant patent for details. This algorithm is used for both self-regulating and integrating loops.

Purpose
Detection of valve stiction.

Interpretation
If the algorithm finds indications for stiction, the index returns +1, if the algorithms find indications for either external (oscillating) disturbances or internal loop problems (most likely too tight a controller tuning), the index returns 0. If no safe decision can be made, the index returns the value –1.

Unit & range
Flag [0,1], = -1 if no safety decision can be taken.

Nominal
No stiction = 0.

Pre-requisite
Loop has been detected as oscillating.

Background
Identify regular abrupt changes in the control error signal.

Reference
Patent „A METHOD AND A SYSTEM FOR EVALUATION OF STATIC FRICTION", WO002002039201A1

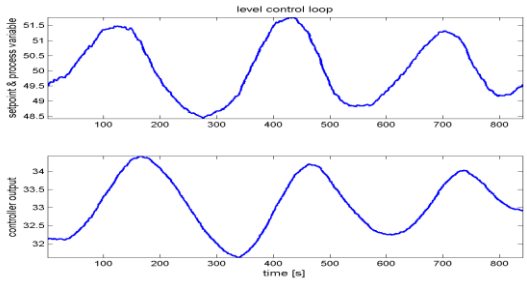
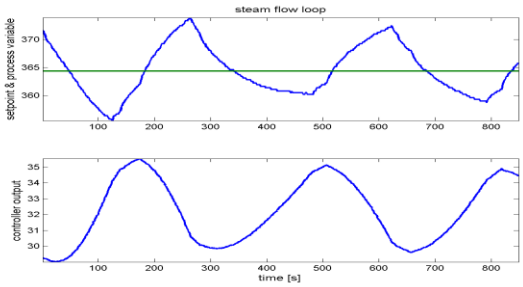
Good Situation	Bad Situation
	
Index = 0	Index = 1

ABB. All rights reserved

12.30 *Valve travel per hour*

General data
LPM Name: valve_travel
Category: Valve

Description
It is highly desirable to find a means of detecting loops where the valve is moving around a lot, as this will lead to excessive mechanical wear. This can be inferred from the PID output signal. It is not as simple as integrating the output signal, as very small output moves will not be sufficient to move the valve. The estimated noise level is used to distinguish noise from real valve movement.
The final 'distance' traveled is normalized to a time base of one hour.

Purpose
Quantify valve actuation effort.

Interpretation
The higher the value, the more the valve has to work, the more aggressively the controller is tuned.

Unit & range
[0, inf]% of Loop Range per hour (%LR/h).

Nominal
Depending on the loop type, the number should be low.

Pre-requisite
None

Background
None

Reference
None

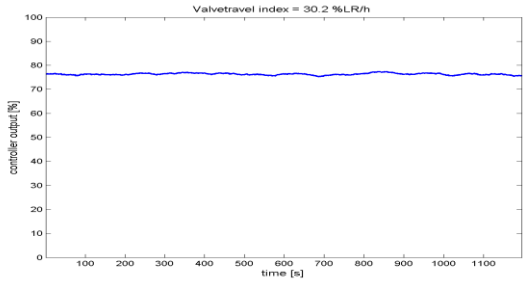
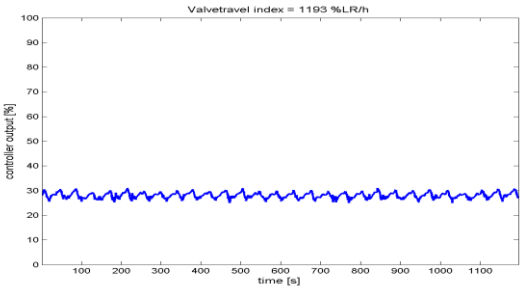
Good Situation	Bad Situation
	
Index = 30.2 %LR/h	Index = 1193 %LR/h

ABB. All rights reserved

12.31 *Valve reversal per hour*

General data
LPM Name: valve_reversals
Category: Valve

Description
It is highly desirable to find a means of detecting loops where the valve is moving around a lot, as this will lead to excessive mechanical wear. This can be inferred from the PID output signal. This index counts the number of valve direction changes during a data batch. The estimated noise level is used to distinguish noise from real valve movement. The final number of reversals is normalized to a time base of one hour.

Purpose
Quantify valve actuation effort.

Interpretation
The higher the value, the more the valve has to work, the more aggressively the controller is tuned.

Unit & range
[0, inf] reversals per hour.

Nominal
Depending on the loop type, the number should be low.

Pre-requisite
None

Background
None

Reference
None

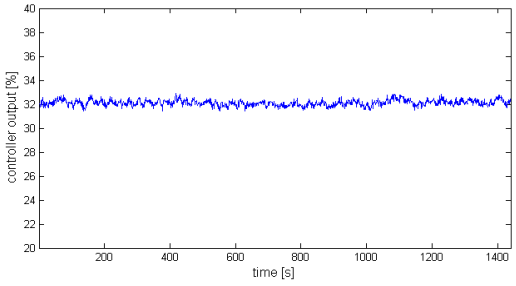
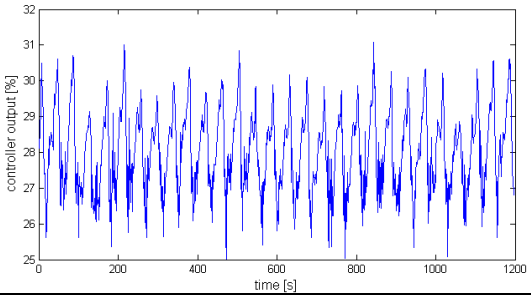
Good Situation	Bad Situation
	
Index = 132 #/h	Index = 912 #/h

ABB. All rights reserved

12.32 *Valve sizing Index*

General data
LPM Name: valve_dimension
Category: Valve

Description
The valve sizing index quantifies the amount of time where the valve works in its normal operating range (between 10% and 90%). The analysis is made on the controller output signal.

Purpose
Identify badly dimensioned valves, i.e. valves that work for a significant amount of time in an inefficient operating range.

Interpretation
The index quantifies the amount of time per batch where the valve is operated in the range below 10% or above 90%.

Unit & range
[0...100] %

Nominal
Nominally, the valve should not be in the low or high range more than 50% of the time, i.e. the index should be larger than 50%.

Pre-requisite
None

Background
A properly dimensioned valve should operate in the range 10-90% most of the time.

Reference
None

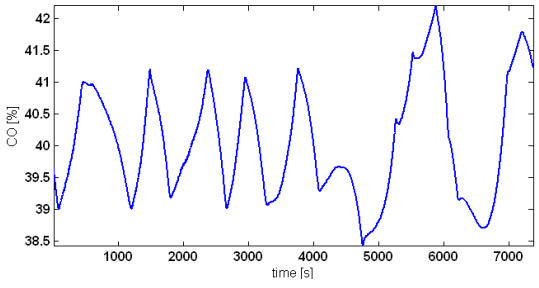
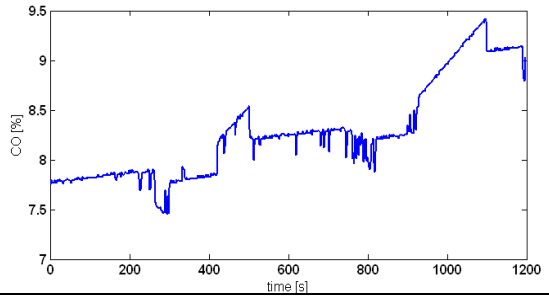
Good Situation	Bad Situation
	
Index = 100%	Index = 0%

ABB. All rights reserved

12.33 *Non-linearity 1st Index*

General data
LPM Name: nonLin_Index_1
Category: Non-linearity

Description
This index quantifies the amount of nonlinearity within a signal. It is used to characterize oscillating signals only.

Purpose
Nonlinearity detection for root-cause analysis of bad control performance.

Interpretation
A signal is considered to be linear if the index is below 1. The higher the index is, the more significant the nonlinearity can be considered.

Unit & range
[0, inf], dimensionless

Nominal
A signal is considered to be linear if the index is below 1.

Pre-requisite
Oscillating control loop

Background
Nonlinearity detection using higher-order statistics.

Reference
T. Schreiber: "Efficient neighbor searching in nonlinear time series analysis." In: Int. Journal of Bifurc. & Chaos 5, 349 (1995).

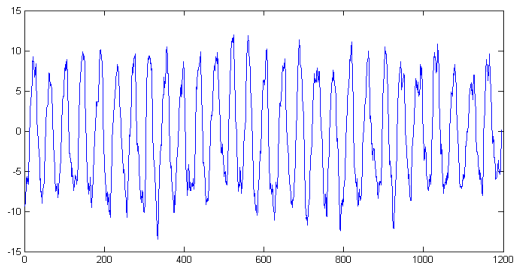
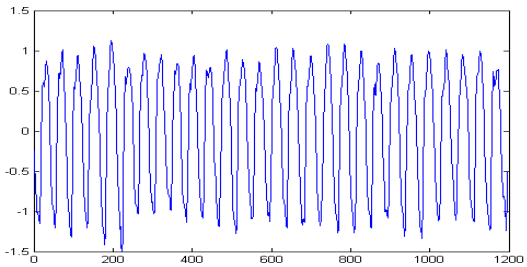
Good Situation	Bad Situation
	
Index = 0.51 (Linear)	Index = 1.63 (non-linear)

ABB. All rights reserved

12.34 *Non-linearity 2nd Index*

General data
LPM Name: nonLin_Index_2
Category: Non-linearity

Description
This index quantifies the amount of nonlinearity within a signal. It is based on bispectrum analysis using higher-order statistics.

Purpose
Nonlinearity detection for root-cause analysis of bad control performance.

Interpretation
A signal is considered to be linear if the index is below 0.15. The higher the index is, the more significant the nonlinearity can be considered.

Unit & range
[0, inf], dimensionless

Nominal
A signal is considered to be linear if the index is below 0.15.

Pre-requisite
Closed-loop, sufficiently long data set.

Background
Nonlinearity detection and higher-order statistics.

Reference
Choudhury, M. A. A. S., Shah, S. L. and Thornhill, N. F., "Diagnosis of Poor Control Loop Performance using Higher Order Statistics", Automatica, **40** 1719-1728, October 2004.

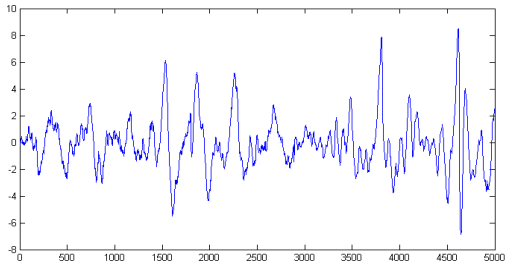
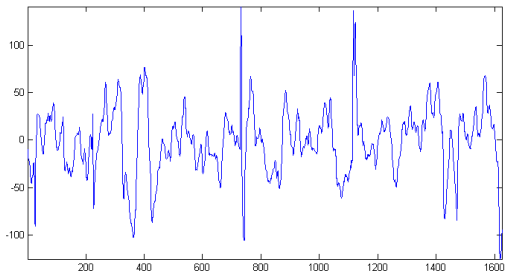
Good Situation	Bad Situation
	
Index = 0.12 (Linear)	Index = 19.2 (non-linear)

ABB. All rights reserved

12.35 *Non-Gaussianity Index*

General data
LPM Name: nonGauss_Index
Category: Non-linearity

Description
This index quantifies the amount of non-Gaussianity within a signal. It is based on bispectrum analysis using higher-order statistics.

Purpose
If a signal is Gaussian, no nonlinearity has to be considered.

Interpretation
The signal is considered to be non-Gaussian if the index is above 0.01.

Unit & range
Real number, dimensionless

Nominal
Non-Gaussian signals should be below 0.01.

Pre-requisite
Closed-loop, sufficient amount of data.

Background
Nonlinearity detection and higher-order statistics.

Reference
Choudhury, M. A. A. S., Shah, S. L. and Thornhill, N. F., "Diganosis of Poor Control Loop Performance using Higher Order Statistics", Automatica, **40** 1719-1728, October 2004.

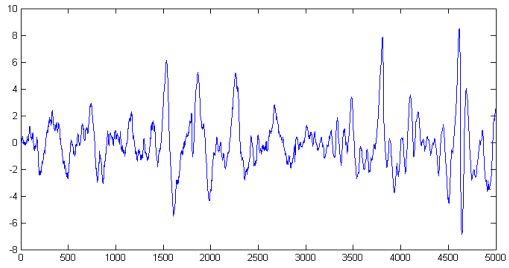
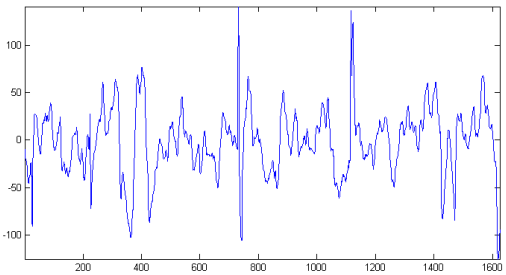
Gaussian	Non-Gaussian
	
Index = -0.21	Index = 4.82

ABB. All rights reserved

12.36 *Oscillation Symmetry Index*

General data
LPM Name: osc_symmetry
Category: Non-linearity

Description
The index verifies if first half period is similar to second half period. This is performed by looking for if the areas under the first half of the half-cycle (left of the maximum) is equal to the area under the second half of the half cycle (right of the maximum)

Purpose
This index describes the symmetry of the oscillation half-periods. A non-symmetric oscillation is an indication for nonlinear behavior.

Interpretation
The closer the index is to 100% the more symmetric the half periods are. The closer to 0, the more skew are they.

Unit & range
[0...100] %

Nominal
The oscillation is considered symmetric if the index is greater than 40%.

Pre-requisite
Detected oscillation.

Background
The reasoning behind this measure is the fact that oscillation often exhibits a non-symmetric behavior, if they are due to non-linearity.

Reference
None

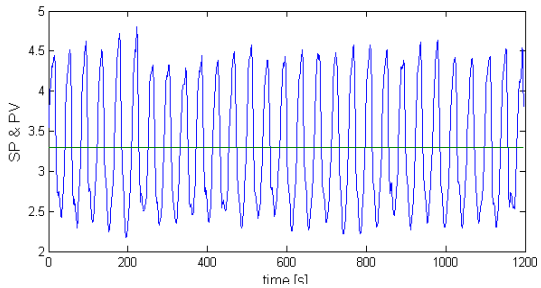
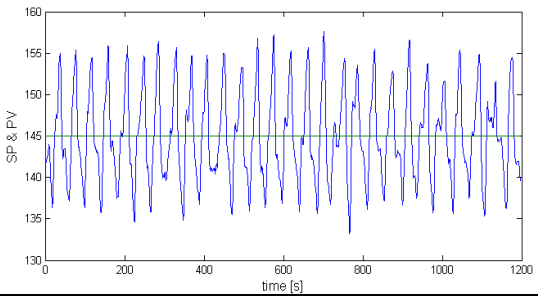
Symmetrical	Asymmetrical
	
Index = 81%	Index = 35%

ABB. All rights reserved

12.37 *Estimated process time-delay*

General data
LPM Name: estimated-delay
Category: Property

Description
If the data batch contains data that enables automatic estimation of the process time delay, estimation is performed and the estimated delay is stored. This value can be used for tuning and may facilitate tuning efforts.

Purpose
Store delay for tuning procedure.

Interpretation
Apparent process time delay in seconds.

Unit & range
seconds

Nominal
Not applicable

Pre-requisite
Data needs to be sufficiently exciting and automatic estimation must be successful.

Background
Theory on time-delay estimation.

Reference
S. Björklund, L.Ljung. "A Review of Time-Delay Estimation Techniques", in 42nd IEEE Conference on Decision and Control", Maui, Hawaii, US, 2003.

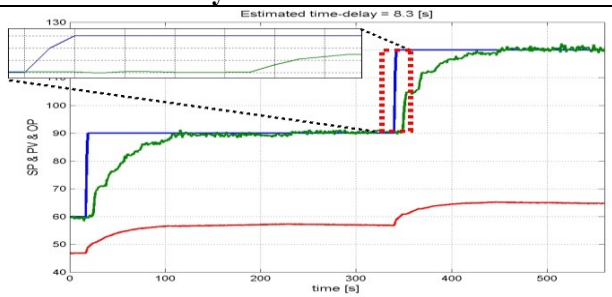
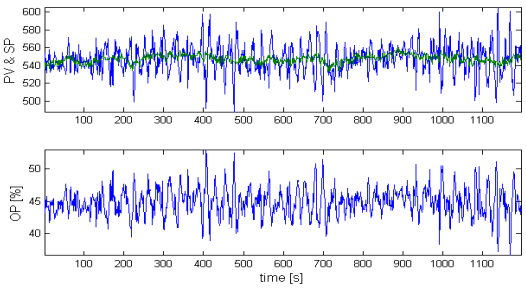
Delay Can Be Estimated	Delay Cannot Be Estimated
	
Index = 8.3 [s]	Index = -9999

ABB. All rights reserved

12.38 *Master Flag*

General data
LPM Name: flag_master_slave
Category: Property

Description
This flag documents whether a loop is the inner or outer loop in a cascade. This flag is for documentation and determines the prediction horizon used in the Harris performance index.

Purpose
To document and store information on the loop configuration.

Interpretation
This index is ‚True' (1) is the loop is the outer loop (‚master') in a cascade. The value is ‚False' (0) if the loop is the inner part of a cascade (‚slave') or a single control loop. This is the default.

Unit & range
[0, 1]

Nominal
Index =0 ⇒ Inner loop in a cascade control or Single Loop
Index =1 ⇒ Outer loop in a cascade control

Pre-requisite
None.

Background
The performance benchmark on the outer cascade loop should be different from the inner loop benchmark.

Reference
None

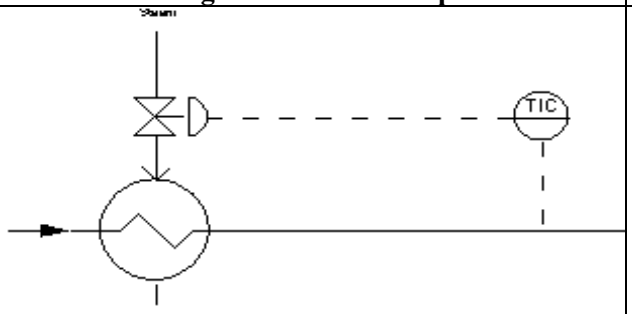
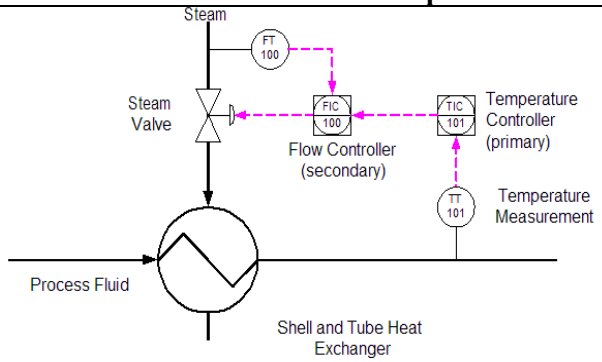
Single or Internal Loop	Master-cascade Loop
	
Index = 0	Index = 1

ABB. All rights reserved

12.39 *Setpoint step flag*

General data
LPM Name: flag_setpoint_step
Category: Property

Description
This index is a flag for data sets that contain setpoint steps.

Purpose
Store information that is useful for tuning. Scanning historical data for useful tuning and identification data can be omitted in this way.

Interpretation
If the index is 1, then one or several setpoint steps in the data were identified. Otherwise the index is 0.

Unit & range
[0; 1]

Nominal
0= no setpoint steps. 1= steps in the setpoint signal

Pre-requisite
None

Background
None

Reference
None

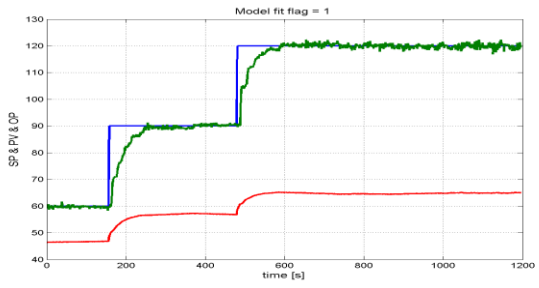
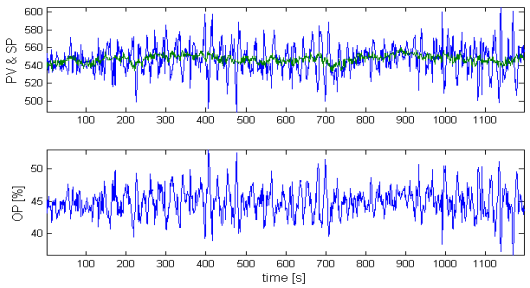
Steps	No steps
	
Index = 1	Index = 0

ABB. All rights reserved

12.40 *Model fit flag*

General data
LPM Name: flag_model_fit
Category: Property

Description
This index is a flag for data sets that are potentially suitable for model identification. The flag is raised if model identification can be performed with an acceptable amount of confidentiality.

Purpose
Identify data sets that can be used for tuning. This saves the user from performing tuning experiments.

Interpretation
If the index is 1, then the data set can be used for model identification. Otherwise the index is 0.

Unit & range
[0; 1]

Nominal
0= data cannot be used for model identification. 1= data can be used also for model identification

Pre-requisite
None

Background
None

Reference
None

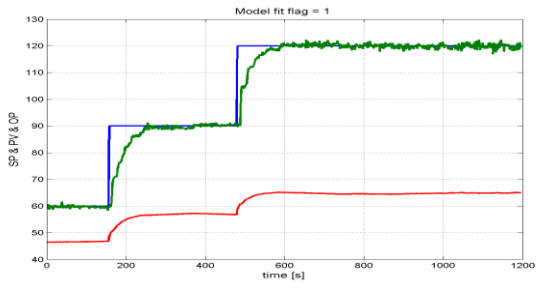
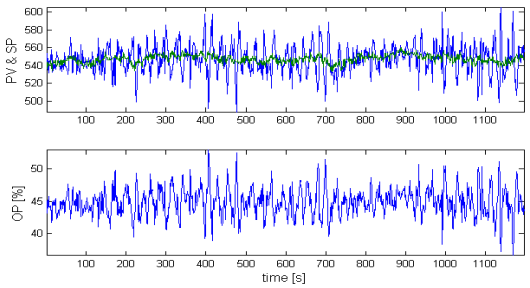
Model Can Be Estimated	Model Cannot Be Estimated
	
Index = 1	Index = 0

ABB. All rights reserved

12.41 *Static Gain Index*

General data
LPM Name: static_gain
Category: Property

Description
Since auditing scans large amounts of data, all useful information from such data sets should be extracted and stored. One such information that is very relevant for tuning is the static input output behavior of a loop. For each data set, one point in such a map can be stored.

Purpose
Store static input / output gain for tuning purposes, e.g. static nonlinearity analysis.

Interpretation
The index contains two values. The digits before the comma denote the static PV value [engineering units], the digits after the comma denote the static Controller Output value [%]. The example on the right shows how to interpret the index.

Unit & range
Not applicable

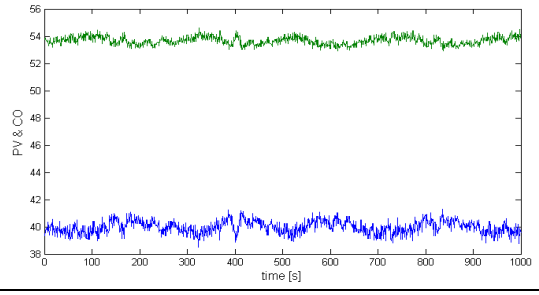
Nominal
Not applicable

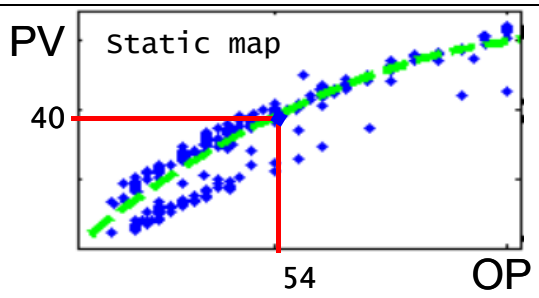
Pre-requisite
None

Background
None

Reference
None

Example





Index = 40.54 ⇒ OP=54, PV=40

ABB. All rights reserved

12.42 *Prediction horizon*

General data
LPM Name: horizon_cont_perf
Category: Housekeeping

Description
This index stores the prediction horizon that was used for the specific loop category for the Harris index (see also there).

Purpose
Store prediction horizon for analysis.

Interpretation
The horizon denotes the time [s] that the benchmark controller would need to remove a disturbance.

Unit & range
Positive values in seconds.

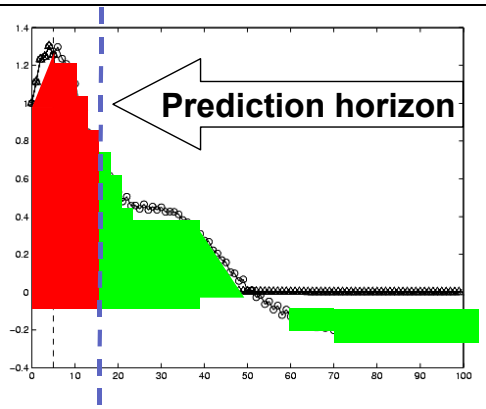
Nominal
Not applicable

Pre-requisite
None

Background
See description of Harris index (index no. 20).

Reference
Thornhill, N.F., Oettinger, M., and Fedenczuk, P., 1999, Refinery-wide control loop performance assessment, *Journal of Process Control*, 9, 109-124

Example



Prediction horizon definition

loop type	master loop	slave or single loop
flow (liquid)	30	10
flow (steam /gas)	30	30
temperature	200	100
composition	30	10
pressure	100	50
level tight	100	50
level average	200	100
other SRP	100	50
other INT	100	50

Default Prediction horizon in LPM

ABB. All rights reserved

12.43 *Loop Category*

General data LPM Name: loop_category Category: Housekeeping
Description This index reports on which „Loop Category“the loop was assigned.
Purpose Reflect different requirements for different types of control loops for performance benchmarking.
Interpretation None
Unit & range Not applicable
Nominal Not applicable
Pre-requisite None.
Background See description of Harris index (index no. 20).
Reference None.

ABB. All rights reserved

12.44 *Area Tag*

General data LPM Name: AreaTag Category: Housekeeping
Description LPM tracks in which working condition the data collection was performed. This is obtained via a special tag for every Area defined as “Classification Tag”, which define the Working Condition Class. This index reports for the Working Condition Class, when the data collection was performed.
Purpose This index is useful for Batch Process or in case of different process behavior for different working load.
Interpretation None.
Unit & range [1, 2, 3, 4, 5]
Nominal Not applicable
Pre-requisite Area classification tag defined
Background None.
Reference None.

ABB. All rights reserved

12.45 *User Defined Indices*

General data LPM Name: user_index_1, user_index_2, user_index_3 Category: User index
Description LPM allows to define 3 Indices defined by user.
Purpose None
Interpretation None
Unit & range Not applicable
Nominal Not applicable
Pre-requisite None
Background None
Reference None

ABB. All rights reserved

12.46 *Automatic Mode Index (Continuous)*

General data
LPM Name: AutoMode
Category: Continuous Indices

Description
Based on the controller output, this index quantifies the amount of time the control loop is in automatic mode. The index checks the variability of the controller output.

Purpose
Track service factor of the control loop.

Interpretation
Amount of samples in automatic per data batch

Unit & range
[0...100%] of samples in data batch

Nominal
100%

Pre-requisite
None

Background
This index reads the Auto/Manual tag to identify the amount of time in Auto mode for the loop, expressed

Reference
None

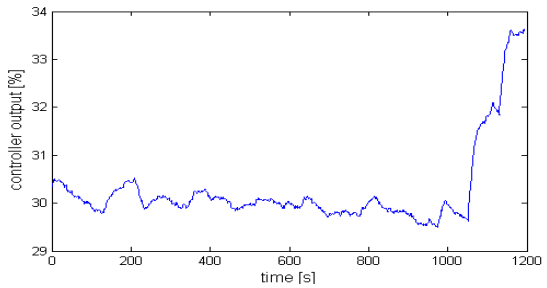
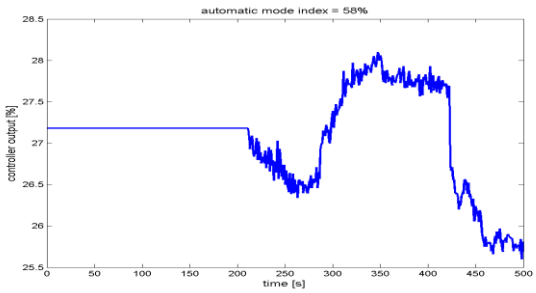
Good Situation	Bad Situation
	
Index = 100%	Index = 56%

ABB. All rights reserved

12.47 *Mean Absolute Loop Error (Continuous)*

General data
LPM Name: MeanAbsoluteError
Category: Continuous Indices

Description
The control error mean is an important measure of control performance. This index calculates the absolute value of control error. It has to be close to zero if no permanent off-set is present.

Purpose
Assess the off-set of the (normalized) control error.

Interpretation
The value of the absolute control error mean is always a positive number and it should be close to zero.

Unit & range
Engineering unit, [0, +max]

Nominal
It is important to see the trend of this index. An increase of the index, compare the past value, should suggest to investigate the loop.

Pre-requisite
None

Background
None

Reference
None

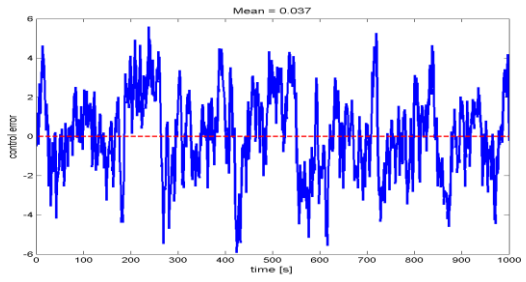
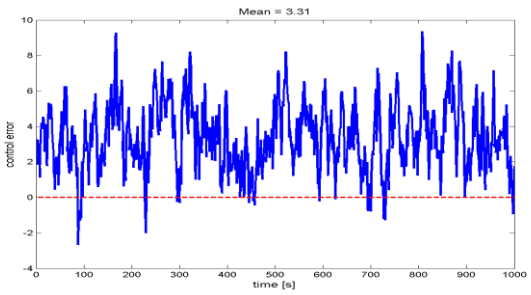
Good Situation	Bad Situation
	
Index = 0.007 t/h	Index = 3.31 t/h

ABB. All rights reserved

12.48 *Saturated Percent (Continuous)*

General data
LPM Name: SaturatedPercent
Category: Continuous Indices

Description
The index quantifies times where the controller output stays significantly long below 3% or above 97% of the controller output range.

Purpose
Quantify amount of time the loop is in saturation.

Interpretation
Percentage of saturation in control loop.

Unit & range
[0...100%] of samples in data batch.

Nominal
0%,

Pre-requisite
None

Background
None

Reference
None

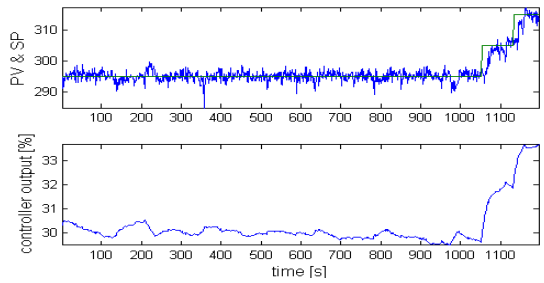
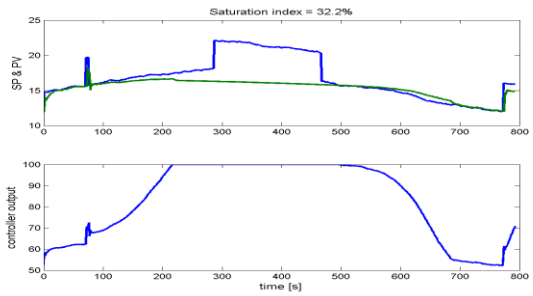
Good Situation	Bad Situation
	
Index = 0%	Index = 32.2%

ABB. All rights reserved

12.49 *Summary Table*

No.	Index	index class	always available?	pre-requisite	unit	min	max	nominal	low limit	up limit	brief description
1	**Mean loop error**	basic statistics	yes	none	real	-Max	Max	0	n.a.	n.a.	control error, SP-PV
2	**Std loop error (norm)**	basic statistics	yes	none	%	0%	100%	0%	0%	5%	standard deviation of normalized control error
3	**Std loop error (eng.unit)**	basic statistics	yes	none	eng. unit	Min	Max	0	0	n.a.	standard deviation of control error in engineering units
4	**Mean PV**	basic statistics	yes	none	eng. unit	Min	Max	n.a.	n.a.	n.a.	mean process variable, current operating point
5	**Std CO**	basic statistics	yes	none	%	0%	100%	n.a.	n.a.	n.a.	standard deviation of controller output
6	**Skewness control error**	basic statistics	yes	none	real	-Max	Max	0	-1	1	measure of signal symmetry
7	**Kurtosis control error**	basic statistics	yes	none	real	-Max	Max	0	-1.5	1.5	measure of signal flatness
8	**Data compression**	Data validity	yes	None	real	0	Inf	0	0	3	Quantify the amount of compression
9	**Data quantization**	Data validity	yes	None	%	0	100%	0%	0%	40%	Verify if data are bad quatizated
10	**Outlier**	Data validity	yes	None	‰	0‰	1000 ‰	0‰	0‰	1‰	Detect and remove outlier
11	**Noise Level**	Data Validity	No	Signal not constant	%	0%	100%	100%	50%	100 %	CE noise level / σ_{CE}
12	**Ratio σCE/σCO**	Loop Modes	yes	None	Real	0	Inf.	n.a.	n.a.	n.a.	Ratio of σ_{CE} and σ_{CO}
13	**Automatic mode**	Loop Modes	yes	none	%	0%	100%	100%	n.a.	100 %	percentage of samples where loop in automatic
14	**Saturation**	Loop Modes	yes	none	%	0%	100%	0	0	2%	percentage of samples when loop saturated

ABB. All rights reserved

No.	Index	index class	always available?	pre-requisite	unit	min	max	nominal	low limit	up limit	brief description
15	**Cascade mode**	Loop Modes	yes	none	%	0%	100%	n.a.	n.a.	n.a.	percentage of samples when loop acts as slave
16	**ACF/pred.horizon ratio**	Performance	no	N>200, Auto Mode	real	0	1	1	0.5	1	ACF instead of closed loop impulse response
17	**setpoint crossing index**	Performance	yes	none	%	0%	100%	25%	10%	100 %	ratio # of setpoint crossings and all samples
18	**cascade tracking index**	Performance	no	Mode Cascade >30%	%	0%	100%	0%	0%	20%	ratio # of setpoint movements and all samples
19	**Cascade travel ratio**	Performance	no	Mode Cascade >30%	real	0	Inf	1	0.5	1.5	ratio movement of SP and PV
20	**Harris index**	Performance	No	N>200, Auto Mode	real	0	1	1	0.5	1	Harris index
21	**Oscillation index CE**	Oscillation	No	N>200, Auto Mode	real	-1	1	1	0.5	1	oscillation index control error (Index >0.5 ⇒Oscillation)
22	**Oscillation index SP**	Oscillation	No	N>200, Auto Mode	real	-1	1	1	0.5	1	oscillation index setpoint (Index >0.5 ⇒Oscillation)
23	**Oscillation period (time)**	Oscillation	No	N>200 Oscillat. detected	[s]	0	inf	n.a.	n.a.	n.a.	period [s] based on time-domain analysis
24	**Oscillation amplitude (time)**	Oscillation	No	N>200 Oscillat. detected	%	0	inf	n.a.	n.a.	n.a.	amplitude based on time-domain analysis

ABB. All rights reserved

No.	Index	index class	always available?	pre-requisite	unit	min	max	nominal	low limit	up limit	brief description
25	**Oscillation severity**	Oscillation	No	N>200 Oscillat. detected	%	0%	100%	n.a.	40%	100 %	percentage of signal power due to oscillation (Index >40% ⇒ Oscillation)
26	**Oscillation Amplitude Regularity**	Oscillation	No	N>200 Oscillat. detected	%	0%	100%	n.a.	60%	100 %	percentage of signal power due to oscillation (Index >60% ⇒ Oscillation)
27	**Shut-off PV value**	Valve	No	CO=0 for an amount of time	Eng. unit	Min. PV	Max. PV	0%	0%	5%	PV value if OP is saturated on lower limit
28	**Stiction (correlation)**	Valve	No	CE oscillation no SP oscillation	flag	0 no stiction	1 stiction	n.a.	n.a.	n.a.	stiction test based on cross-correlation method =–1 if no decision is possible
29	**Stiction (histogram)**	Valve	No	CE oscillation no SP oscillation	flag	0 no stiction	1 stiction	n.a.	n.a.	n.a.	stiction test based on histogram method =–1 if no decision is possible
30	**Valve travel per hour**	Valve	yes	None	% LR/h	0	inf	Low value	0	n.a.	integrated valve movement per hour (with deadzone)
31	**Valve Reversal per hour**	Valve	yes	None	#/h	0	inf	Low value	0	n.a.	Valve direction changes per hour
32	**Valve Sizing Index**	Valve	yes	None	%	0%	100%	100%	50%	100 %	Time outside saturation divided total time
33	**Non-linearity 1st**	Non linearity	yes	Oscillat. detected	Real	0	Inf	0	0	1	Nonlinearity quantification 1st method

ABB. All rights reserved

No.	Index	index class	always available?	pre-requisite	unit	min	max	nominal	low limit	up limit	brief description
34	**Non-linearity 2nd**	Non linearity	yes	Oscillat. detected	Real	0	Inf	0	0	0.15	Nonlinearity quantification 2nd method
35	**Non-Gaussianity**	Non linearity	yes	Close loop. Many samples	Real	-Inf	Inf	-Inf	-Inf	0.01	Gaussianity test using higher-order statistics
36	**Oscillation symmetry**	Non linearity	yes	Oscillat. detected	%	0%	100%	100%	40%	100 %	Verifies if the oscillation is symmetrical
37	**Estimated time-delay**	Property	no	SP exciting (casc>0), auto>50 %, N>200, saturation<25%, osc.<0.5	sec	0	inf	n.a.	n.a.	n.a.	estimation of time-delay
38	**Master flag**	Property	yes	none	[0/1]	0	1	n.a.	n.a.	n.a.	Ind.=0 sigle or Internal loop Ind.=1 Mater loop in a cascade control
39	**Step flag**	Property	yes	none	[0/1]	0	1	n.a.	n.a.	n.a.	Flag 0= no SP steps 1= SP step
40	**Model fit flag**	Property	yes	None	[0/1]	0	1	n.a.	n.a.	n.a.	Flag 0= data not OK for identification 1= data OK for identification
41	**Static Gain**	Property	yes	None	Real. Real	N.A.	N.A.	N.A.	N.A.	N.A.	Example Index 40.54 40=Mean PV 54=Mean OP
42	**Prediction horizon**	Housekeeping	yes	None	sec	0	Inf	N.A.	N.A.	N.A.	default values for each loop category, stored for documentation only

ABB. All rights reserved

No.	Index	index class	always available?	pre-requisite	unit	min	max	nominal	low limit	up limit	brief description
43	**Loop category**	Housekeeping	yes	None	integer	1	9	n.a.	n.a.	n.a.	number which represents the loop categories set for the loop
44	**Area Tag**	Housekeeping	yes	None	integer	1	5	n.a.	n.a.	n.a.	Tracks for the working condition class
45	**User Index 1**	User Index	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	User defined index
46	**User Index 2**	User Index	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	User defined index
47	**User Index 3**	User Index	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	User defined index
48	**AutoMode**	Continuous Index	yes	none	%	0%	100%	100%	N.A.	N.A.	Ammount of time with loop in Automatic
49	**Mean Abs. CE (Continuous)**	Continuous Index	yes	none	Eng. Unit.	0	Inf	Small value	0	N.A.	Absolute value of CE
50	**Saturated percent**	Continuous Index	yes	none	%	0%	100%	100%	50%	100 %	Time outside saturation divided total time

Table 15 – Summary of Indices

ABB. All rights reserved

13. Appendix E - Description of the Diagnosis Hypothesis

LPM elaborates maintenance hypothesis on possible problems on the basis of the calculated indices. The following chapters present a brief explanation on the meaning of every diagnose suggestion, in order to help the maintenance crew to easily focus on the problems resolution.

13.1 *Acceptable overall performance*

Description

Summarize information from seven different hypotheses for an overall assessment. This overall assessment merely describes that the behavior is not satisfactory. The other diagnoses deal rather with possible underlying root causes.

For each batch evaluation:

$$\mathbf{H_1 = P_1 \wedge P_2 \wedge .... \wedge P_7}$$

For report:

$$\mathbf{H_1 = a_1 \cdot \mu(P_1) \wedge a_2 \cdot \mu(P_2) \wedge .... \wedge a_7 \cdot \mu(P_7)}$$

(where μ is the mean value calculated over all batches)

H1: Excellent (90%), Good (>80%), Fair (>50%), Poor (<50%)

The 7 pre-conditions are shown in Table 6 - LPM Diagnostic Hypothesis and Pre-conditions.

Purpose

Overall assessment based on a single number.

Interpretation

In general the diagnosis suggestions are true or false. Instead for Acceptable Overall Performance possible outputs are: Poor, Fair, Good or Excellent. In this way the maintenance engineers can easily focus on the worst performing loops.

ABB. All rights reserved

13.2 *Tuning problem*

Description Unacceptable controller performance most likely due to bad tuning. This diagnosis is applicable to both oscillating and non-oscillating loops.
Diagnose reasoning Non-oscillating loops: the loop should not exhibit significant nonlinearities and the tuning indices (ACF ratio and Harris index) should both indicate improvement potential. [see Example 1]. Oscillating loops: the loop should not oscillate due to a nonlinearity (stiction and others) and the Harris index should be low and the oscillation amplitude should vary significantly [see Example 2].

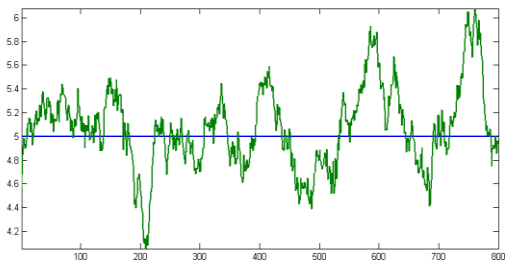
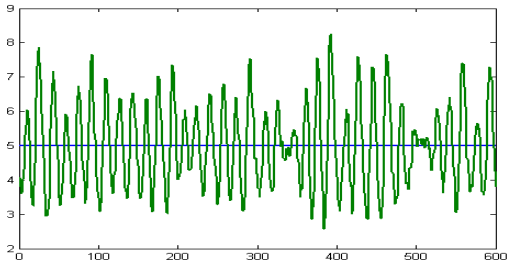
Slow Tuning	Fast Tuning
	

ABB. All rights reserved

13.3 *Loop Oscillatory*

Description Unacceptable controller performance clearly visible by a periodic deviation from the target. Loop oscillations are rather a symptom than a cause but the fact „loop oscillatory“ is an important message to the user.
Diagnose reasoning The oscillation diagnosis is based on the oscillation indices. If an oscillation is detected, its main period and amplitude is described by the different indices. Other relevant measures like symmetry, regularity, severity etc. are important for more detailed analysis. The oscillation diagnosis is not used if the data was quantized.

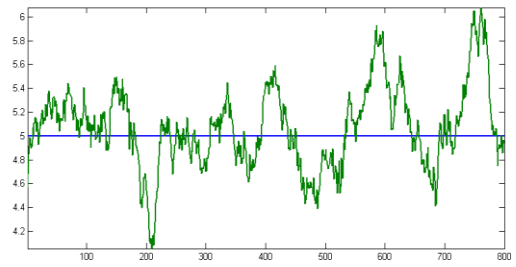
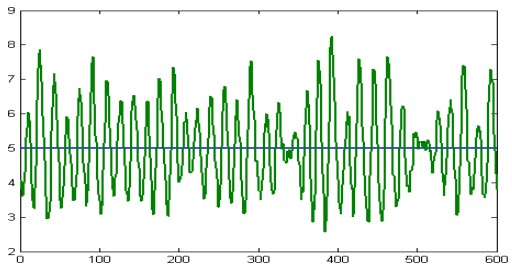
Good situation – Non Oscillatory	Bad Situation - Oscillatory
	

ABB. All rights reserved

13.4 *Setpoint Oscillatory*

Description A loop may behave well, even though it is oscillating. One reason for this may be an oscillatory setpoint. This diagnoses checks if this is a likely cause.
Diagnose reasoning The analysis is based on the oscillation index applied to the setpoint signal

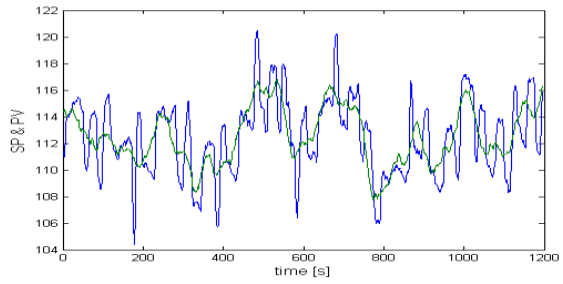
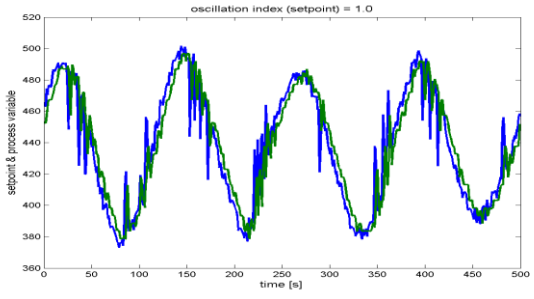
Good situation – SP Non Oscillatory	Bad Situation – SP Is Oscillating
	

ABB. All rights reserved

13.5 *Significant external disturbance*

Description A loop may be oscillating due to a periodic external disturbance. The controller is not able to remove this oscillation, either because it was not designed to do so or because it is insufficiently tuned for that task.
Diagnose reasoning If the controller is in manual mode, a significant oscillation is always externally generated. If the controller is in automatic mode, a signficant external oscillation is identified based on the following facts: The oscillation is rather linear, no stiction was detected, the oscillation is rather symmetric and the loop is not saturated

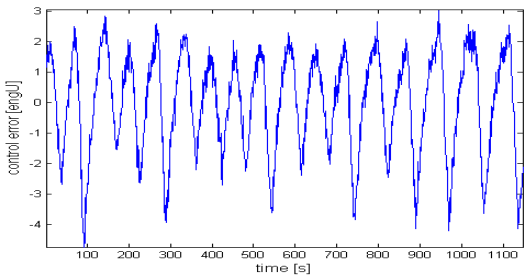
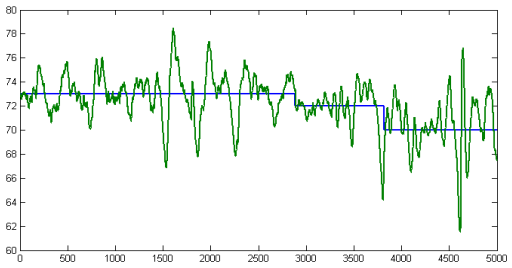
Internally Generated – No Disturbance	Externally Generated - Disturbance
	

ABB. All rights reserved

13.6 *Significant non linearity*

Description

Many problems in control loop performance are caused by significant nonlinearity within the control loop. This diagnosis indicates such problems

Diagnose reasoning

If the controller is in manual mode, a significant oscillation is always externally generated.

If the controller is in automatic mode, a signficant external oscillation is identified based on the following facts: The oscillation is rather linear, no stiction was detected, the oscillation is rather symmetric and the loop is not saturated.

The analysis is based on nonlinearity analysis involving several indices

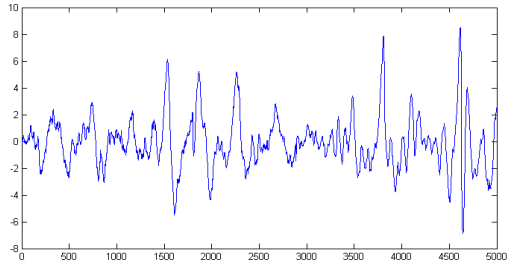
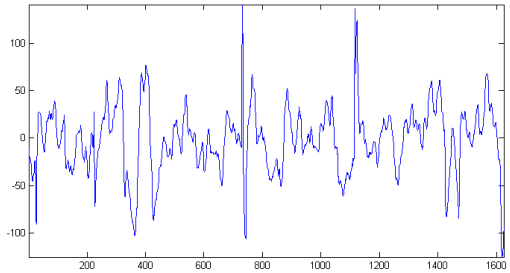
Linear variation	Non-linear variation
	

ABB. All rights reserved

13.7 *Valve stiction*

Description

Valve stiction is present in many control loops that involve control valves. The diagnosis valve stiction is mainly related to clear oscillations in the control loop.
In order to identify stiction without oscillations present, typically, setpoint changes have to be performed manually.

Diagnose reasoning

Valve stiction diagnosis involves quite a few indices, among them automatic mode, nonlinearity analysis and stiction indices. This is necessary since stiction is based on several pre-conditions and to avoid false alarms as much as possible

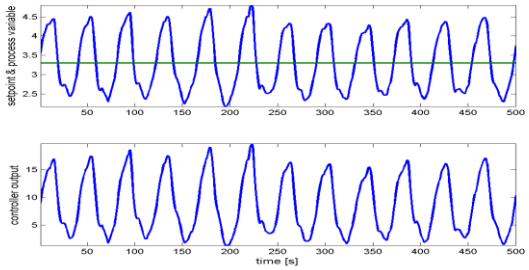
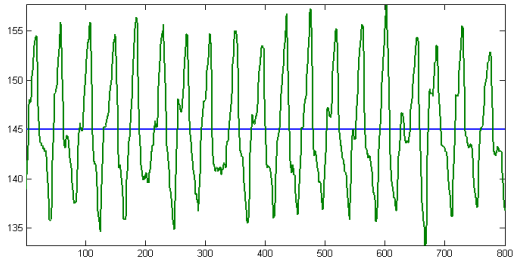
No valve stiction	Valve stiction
	

ABB. All rights reserved

13.8 *Valve leakage*

Description This diagnosis checks for valve leakage or instrument zero errors. If the controller forces the output to be zero and the PV does not follow this requirement, then either the valve is not completely shut or the instrument cannot measure zero flow
Diagnose reasoning The diagnosis is based on the assessment of the resulting PV value when the control valve is required to be shut for a certain while.

Valve leakage

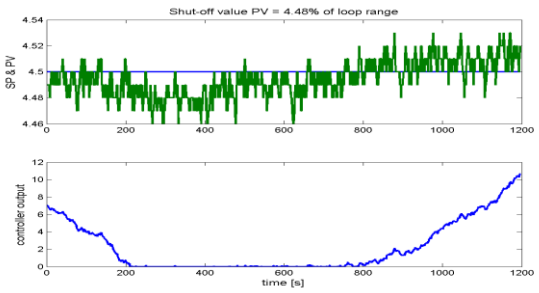


ABB. All rights reserved

13.9 *Valve size incorrect*

Description

Valves are usually designed to operate outside the extreme levels of open or closed. If a valve operates within these ranges a longer period of time, this can be an indication for wrongly dimensioned valve, thus possibly resulting in non-optimal loop performance.

Diagnose reasoning

The diagnosis is based on the estimation of time where the valve operates in the extreme positions close to 100 and 0%. This value is delivered by the valve dimension index.

Valve size incorrect

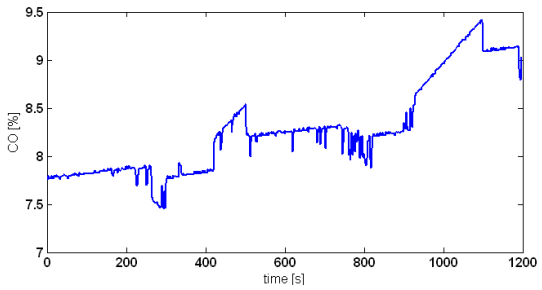


ABB. All rights reserved

13.10 *Excessive valve action*

Description

Even if the controller performance is good, it can be on cost of the valve lifetime since the valve might have to work excessively. This diagnosis indicates excessive valve movement, no matter what the overall performance is (can be good or bad).

Diagnose reasoning

The diagnosis is based on the corresponding indices dealing with valve movement. An important pre-requisite is the handling of noise, which is not related to valve movement.

Excessive valve action

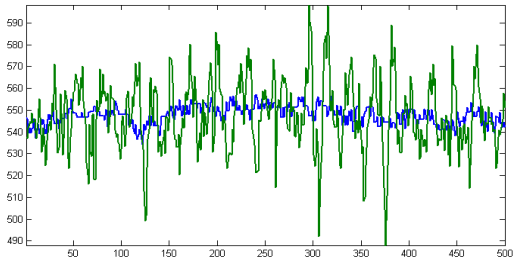


ABB. All rights reserved

13.11 *Data unreliable*

Description Any signal-based analysis can be misleading by using data that is not suitable for such analysis. Tests for the suitability of the current data batch form this diagnosis.
Diagnose reasoning Especially data that has been compressed prior to analysis can lead to unreliable results. The data is tested on outliers (that are removed automatically), compression and quantization.

Data unrealible

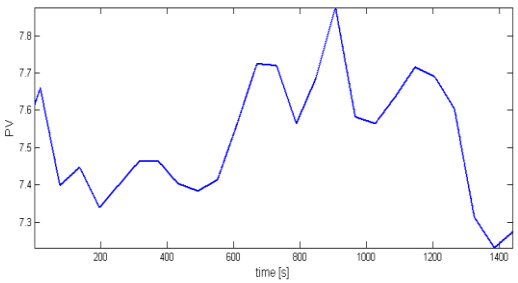


ABB. All rights reserved

14. Appendix F: PID Controller Types

During loop configuration, the user should select a specific PID controller type, e.g. ABB BAILEY FC156 S18=0,S20=0. The default PID type is the parallel form, however LPM includes many other industrial PID controller types. This section discusses the options included for each of the generic PID types and for each specific industrial PID type.

14.1 *Background*

The most general expression for the PID controller algorithm is the parallel form. When allowing for setpoint weighting in the proportional part, and for derivative action on either the process value or the process deviation, the PID equation can be written as follows:

$$u = K_P(\beta r - y) + K_I \frac{1}{s}(r - y) + K_D \frac{s}{1 + T_F s}(\gamma r - y)$$

where

u	Manipulated variable value
r	Setpoint
y	Process variable value
K_P	Proportional gain
K_I	Integral gain
K_D	Derivative gain
T_F	Filter time constant
β	Setpoint weight in proportional part
γ	Setpoint weight in derivative part

Controllers designed for “Proportional on PV” are represented by β=0 and controllers designed for “Proportional on deviation” are represented by β=1. Some PID types allow the beta setpoint weight to be adjustable.

Controllers designed for “Derivative on PV” are represented by γ=0 and controllers designed for “Derivative on deviation” are represented by γ=1.

It is common that the values of the gains are given as positive numbers and to use a separate configuration parameter to specify direct action (as opposed to the ‘standard’ reverse action), when negative values are desired. Negative gains (direct action) will be desired, when the static gain of the process (from manipulated variable to process variable) is negative.

Most PID controllers also include some form of scaling to normalize the equations and the resulting gain parameters. Many controllers use the PV range and the CO range for scaling. Some controllers include a separate scaling parameter.

ABB. All rights reserved

14.2 *Generic PID forms*

LPM supports three generic PID forms: parallel, ideal, and series. Each of these forms is described below.

14.2.1 PARALLEL

The generic parallel form in LPM is based on the following equation:

$$u = K_P(\beta r - y) + K_I \frac{1}{s}(r - y) + K_D \frac{s}{1 + T_F s}(-y)$$

Note that this form assumes "derivative on PV" and the beta setpoint weight is adjustable. No scaling is used.

14.2.2 IDEAL

The ideal form is based on the following equation:

$$u = K\left[\beta r - y + \frac{1}{T_I s}(r - y) + \frac{T_D s}{1 + T_F s}(-y)\right]$$

This form also assumes "derivative on PV" and the beta setpoint weight is adjustable. No scaling is used.

14.2.3 SERIES

The series form is based on the following equation:

$$u = K\left[\beta r - y + \frac{1}{T_I s}\left(r - \frac{1 + T_D s}{1 + T_F s} y\right)\right]$$

This form also assumes "derivative on PV" and the beta setpoint weight is adjustable. No scaling is used.

ABB. All rights reserved

14.3 *Industrial PID forms*

LPM also supports several industrial PID controller types:

ABB AC800 Series
ABB Bailey
- ABB Hartmann and Braun Contronic P
- ABB Advant/Master
- ABB Mod300
- ABB AlfaLaval Control

Honeywell TDC

Each industrial PID controller type includes various subtypes, which are defined when there are different algorithms (e.g. different function codes, PI vs. PID, etc.) or different configuration options (e.g. derivative on PV vs. derivative on error, proportional on PV vs. proportional on error, etc.).

The following list describes each of the PID controller subtypes included in LPM.

14.3.1 ABB AC800 Series

14.3.1.1 AC800 F

The AC800F PID controllers allow the user to configure the time units to seconds, minutes, or hours. However, in LPM, the time units are always assumed to be in *seconds* for the reset time and derivative time. Therefore, the user must convert the tuning results to minutes or hours if the actual PID controller is configured for such time units. The PV range and CO range are used for scaling. The derivative constant CD should be between 0 and 20 and the default is 10.

ABB AC800F D OF CE, P OF CE
This form assumes derivative on control error and proportional on control error.

ABB AC800F D OF PV, P OF CE
This form assumes derivative on PV and proportional on control error.

ABB AC800F D OF PV, P OF PV
This form assumes derivative on PV and proportional on PV.

14.3.1.2 AC800 M/C

The PV range and CO range are used for scaling. The integral time and derivative time are both defined in seconds.

ABB AC800M/C BETA=0
This form assumes derivative on PV and proportional on PV.

ABB AC800M/C BETA=1
This form assumes derivative on PV and proportional on error.

ABB. All rights reserved

14.3.2 ABB Bailey

14.3.2.1 FC18 and FC19

The user should set the overall gain constant K (S5) to convert the signal to a percentage scale. The derivative constant is defined in minutes and the integral constant is defined in 1/minute. The LPM tuning also calculates a value for the derivative (filter) lag constant. This value cannot be set individually for each PID, only for each controller module. So if the value is fixed for a PID, then copy the tuning parameter results and set the derivative lag constant equal to the fixed value and click Calculate to view the correct simulation and frequency response results.

ABB BAILEY FC18
This form assumes derivative on error and proportional on error.

ABB BAILEY FC19 S11=0
This form assumes derivative on error and proportional on error.

ABB BAILEY FC19 S11=1
This form assumes derivative on error and proportional on PV.

14.3.2.2 FC156

The user should set the overall gain constant K (S11) to convert the signal to a percentage scale. The derivative rate action is provided in minutes and the integral reset is in 1/minute. The derivative (filter) lag constant can be adjusted and has a default value of 10.

ABB BAILEY FC156 S18=0,S20=0
This subtype represents the "Classical" PID algorithm and assumes derivative on PV and proportional on error.

ABB BAILEY FC156 S18=0,S20=1
This subtype represents the "Classical" PID algorithm and assumes derivative on PV and proportional on PV.

ABB BAILEY FC156 S18=1,S20=0
This subtype represents the "Noninteracting" PID algorithm and assumes derivative on PV and proportional on error.

ABB BAILEY FC156 S18=1,S20=1
This subtype represents the "Noninteracting" PID algorithm and assumes derivative on PV and proportional on PV.

ABB BAILEY FC156 S18=2,S20=0
This subtype represents the "Classical with external reset" PID algorithm and assumes derivative on PV and proportional on error.

ABB. All rights reserved

ABB BAILEY FC156 S18=2,S20=1
This subtype represents the “Classical with external reset” PID algorithm and assumes derivative on PV and proportional on PV.

14.3.2.3 FC160 – Inferential Smith Controller

The inferential smith controller (ISC) function code provides predictive control on an error signal developed from the process variable and set point inputs measured against an internal model of the process. The ISC controller utilizes a first order dynamic model with deadtime to predict the current value of the process variable based on past values of the control output. The ISC controller function block provides regulatory process control similar to a PID algorithm. However, the ISC controller has the added advantage of effective control for processes with a significant transport delay (deadtime)

For an ISC controller type, the tuning design includes the following restrictions:

- The selected model must be first order plus deadtime (i.e. model can only include a gain, lag time constant, and deadtime).
- The allowable tuning methods are Lambda and Manual tuning.
- The tuning results displayed to user include:
  - S7: Process model gain
  - S8: Process model deadtime (in seconds)
  - S9: Process model lag time constant (in seconds)
  - S10: Controller tuning time constant, i.e. lambda (in seconds)

14.3.2.4 ABB Hartmann and Braun Contronic P

LPM does not apply any scaling for these PID types. The integral action time and derivative action time are defined in minutes.

ABB CONTRONIC P PI
This is a PI controller, so the derivative term is always 0. This form assumes proportional on error.

14.3.2.5 ABB CONTRONIC P PID

This form assumes derivative on error and proportional on error.

ABB CONTRONIC P PI OF XW + D OF X
This form assumes derivative on PV and proportional on error.

ABB CONTRONIC P I OF XW + PD OF X
This form assumes derivative on PV and proportional on PV.

ABB. All rights reserved

14.3.2.6 ABB Melody APID

This controller has different formulation based on the function type (small/medium or large). Also, there is a differentiation based on the use of process variable only or control error in the proportional part.

ABB MELODY APID SMALL/MEDIUM
This form assumes derivative on PV and proportional on error.

ABB MELODY APID LARGE
This form assumes derivative on PV and proportional on error. Different values are available in the upward and downward cases.

ABB MELODY APID SMALL/MEDIUM NO P-JUMP
This form assumes derivative on PV and proportional on PV.

ABB MELODY APID LARGE NO P-JUMP
This form assumes derivative on PV and proportional on PV. Different values are available in the upward and downward cases.

14.3.2.7 ABB Advant/Master

The integral time and derivative time are both defined in seconds.

ABB MASTER PIDCON D ON ERROR
This form assumes derivative on error and proportional on error. The PV range and CO range are used for scaling.

ABB MASTER PIDCON D ON PV
This form assumes derivative on PV and proportional on error. The PV range and CO range are used for scaling.
ABB MASTER PI
This is a PI controller, so the derivative term is always 0. This form assumes proportional on error. This PID type does not include any scaling.

ABB MASTER PIDCONA D ON ERROR
This form assumes derivative on error and the beta setpoint weight is adjustable. The PV range and CO range are used for scaling. The LPM tuning sets the derivative filter time to 0.1*derivative time, but this parameter can be modified by the user.

ABB MASTER PIDCONA D ON PV
This form assumes derivative on PV and the beta setpoint weight is adjustable. The PV range and CO range are used for scaling. The LPM tuning sets the derivative filter time to 0.1*derivative time, but this parameter can be modified by the user.

ABB. All rights reserved

14.3.2.8 ABB Mod300

The PV range and CO range are used for scaling. The derivative action time is defined in minutes and the reset rate is defined in 1/minute. The derivative filter constant is fixed at 0.1 and cannot be changed. The derivative filter constant is fixed and cannot be changed.

ABB MOD300 INTRACT=NO, P ON E, D ON E
This subtype represents the Mod300 “noninteractive” form and assumes proportional on error and derivative on error.

ABB MOD300 INTRACT=NO, P ON E, D ON P
This subtype represents the Mod300 “noninteractive” form and assumes proportional on error and derivative on PV.

ABB MOD300 INTRACT=NO, P ON P, D ON P
This subtype represents the Mod300 “noninteractive” form and assumes proportional on PV and derivative on PV.

ABB MOD300 INTRACT=YES, P ON E, D ON E
This subtype represents the Mod300 “interactive” form and assumes proportional on error and derivative on error.

ABB MOD300 INTRACT=YES, P ON E, D ON P
This subtype represents the Mod300 “interactive” form and assumes proportional on error and derivative on PV.

ABB MOD300 INTRACT=YES, P ON P, D ON P
This subtype represents the Mod300 “interactive” form and assumes proportional on PV and derivative on PV.

14.3.2.9 ABB AlfaLaval Control

The integral time and derivative time are both defined in seconds.

ABB SATTCON 200 BETA=0
This PID type does not include any scaling. This form assumes proportional on PV and derivative on PV.

ABB SATTCON 200 BETA=1
This PID type does not include any scaling. This form assumes proportional on error and derivative on PV.

ABB SATTLINE BETA=0
The PV range and CO range are used for scaling. This form assumes proportional on PV and derivative on PV.

ABB SATTLINE BETA=1

ABB. All rights reserved

The PV range and CO range are used for scaling. This form assumes proportional on error and derivative on PV.

14.3.3 Honeywell TDC

The PV range and CO range are used for scaling. The integral time constant and derivative time constant are both defined in minutes.

HONEYWELL TDC AM/PM IDEAL, A or *HONEYWELL TDC HPM IDEAL, A*
This subtype represents the Honeywell ideal form and assumes derivative on error and proportional on error.

HONEYWELL TDC AM/PM IDEAL, B or *HONEYWELL TDC HPM IDEAL, B*
This subtype represents the Honeywell ideal form and assumes derivative on PV and proportional on error.

HONEYWELL TDC AM/PM IDEAL, C or *HONEYWELL TDC HPM IDEAL, C*
This subtype represents the Honeywell ideal form and assumes derivative on PV and proportional on PV.

HONEYWELL TDC AM/PM REAL, A or *HONEYWELL TDC HPM REAL, A*
This subtype represents the Honeywell real form and assumes derivative on error and proportional on error.

HONEYWELL TDC AM/PM REAL, B or *HONEYWELL TDC HPM REAL, B*
This subtype represents the Honeywell real form and assumes derivative on PV and proportional on error.

HONEYWELL TDC AM/PM REAL, C or *HONEYWELL TDC HPM REAL, C*
This subtype represents the Honeywell real form and assumes derivative on PV and proportional on PV.

14.3.4 Metso DNA

This controller has a unique formulation available.

ABB. All rights reserved

15. Appendix G: Taglist Creator, Infi90

15.1 *Introduction*

Configuration of loops in LPM for Infi90 and Symphony Harmony systems requires the following information:

- Infi90 address for Process Value (PV)
- Infi90 address for Control Output (CO)
- Infi90 address for SetPoint (SP)
- PID function block algorithm type
- PID function block handling of SP changes for integral
- PID function block parameters (gain, proportional, integral and derivative parameters)
- PID function block direction (direct or reverse)

The above listed information can be retrieved from the system configuration; in particular, information can be extracted from both

- Console database (like OIS database) or access database extracted from Composer
- Module configuration files

Taglist creator module extracts automatically all relevant information from DCS files; it creates excel spreadsheets with the proper format that can then be imported into LPM.

15.2 *How to identify Infi90 loops*

15.2.1 First step: identify names and base address for loops

In this step the software scans a console file searching for all configured loops; the search is performed automatically by looking for all Station blocks (FC80); as a result, a list of the module configuration files requested to complete information about the loop is generated. This second step is in fact necessary to complement Station information with PID function block information

This is done in the first tab of the main application window; by clicking on the upper … button a console file is selected (extension .mdb in the case of composer and .dbf in the case of console files). By clicking read file a search of the configured Station function codes is performed**.**
As a result of the search, the list of the relevant configuration files (.cfg) is loaded into the configuration file table.

ABB. All rights reserved

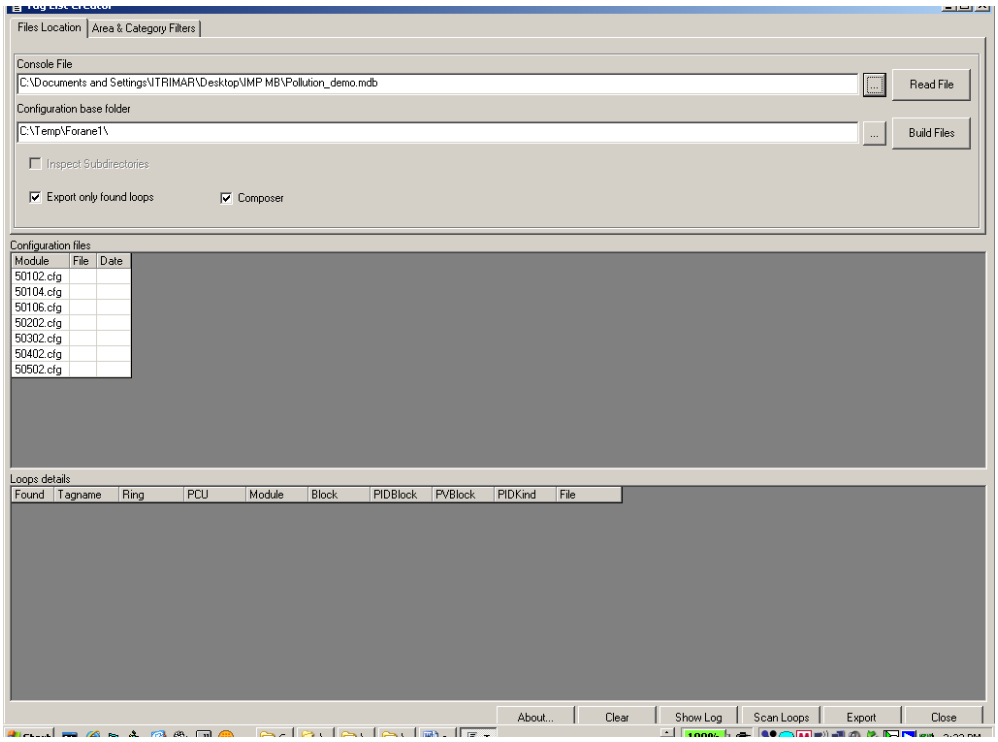


Figure 213 - Taglist creator after database search

Single configuration files can be entered either manually, by clicking on the left column in the grid and by browsing the file name. Symphony Infi90 files should be located and selected in the relevant dialog.

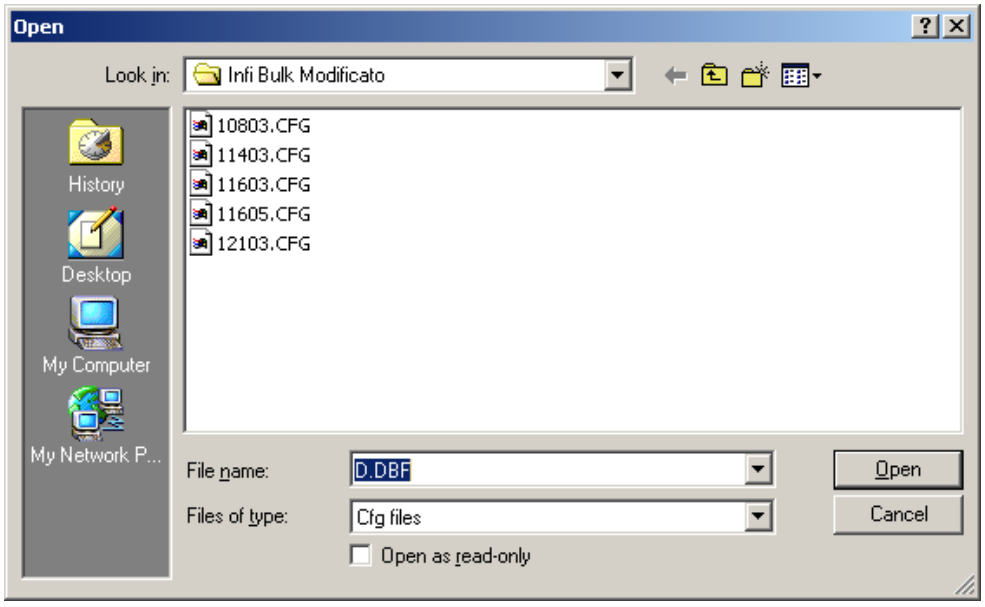


Figure 214 - Manual selection of CFG files

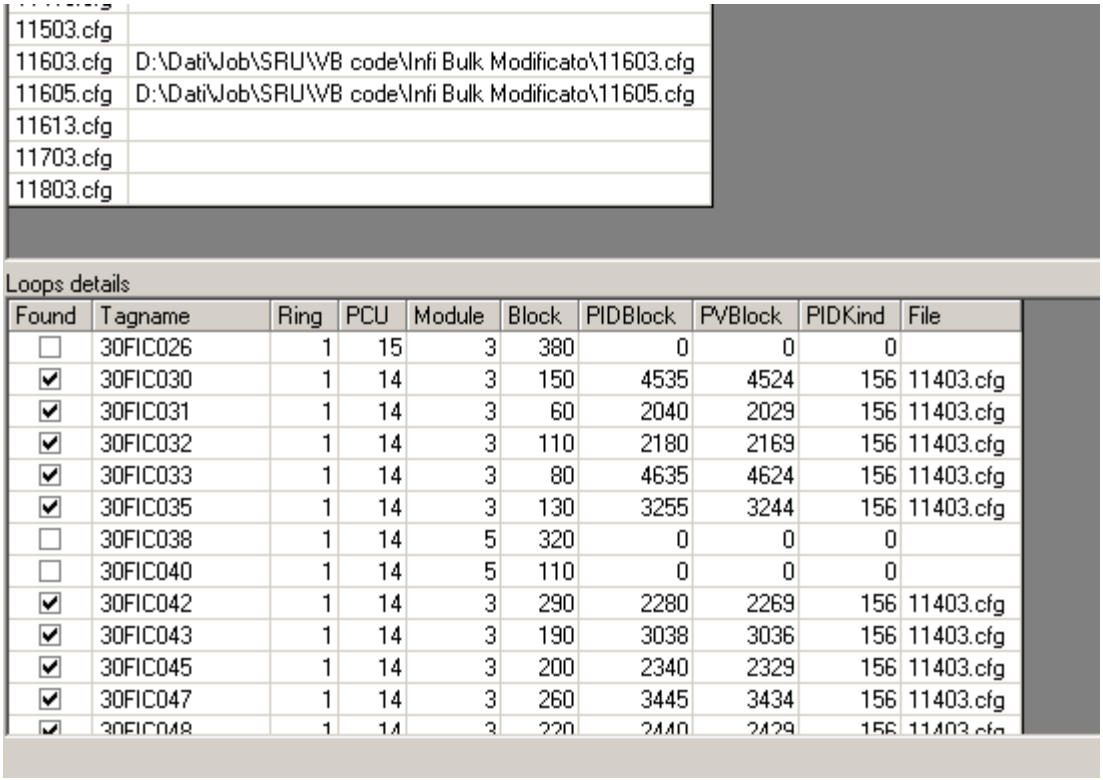
ABB. All rights reserved

As an alternative, if all files are located into a specific directory, the user may specify it by clicking on the lower … button and then by browsing the directory name. Then, by clicking the Read File button, all relevant configuration file names are automatically loaded into the grid.
If the files are organized in a tree-like structure, as for Wincad projects, by selecting the root directory and by clicking the **Inspect Subdirectories** checkbox, the software will automatically look in the proper subfolders to locate the relevant configuration files.

In case of Composer use, by clicking the dedicated checkbox and pressing the read files, the software will automatically search all CFG files located in the same directory as the access tag database file. Il will also sort multiple CFG files present for the same physical module so to select the latest version based on the compile/download date and time.

Once the list is loaded with all the available configuration files and the **Scan Loops** button is clicked, the loops detail table is filled in with the available information for the loops present in the console database and configured in the specified module configuration files.

If any of the configuration file path is left empty, the application will be unable to determine some of the loop addresses and specifications ie. the PID parameters for those loops. If that is the case, the loops will be marked as not found in the table and will not be exported to Excel in the last step unless selected by the user.



11503.cfg	
11603.cfg	D:\Dati\Job\SRU\VB code\Infi Bulk Modificato\11603.cfg
11605.cfg	D:\Dati\Job\SRU\VB code\Infi Bulk Modificato\11605.cfg
11613.cfg	
11703.cfg	
11803.cfg	

Loops details

Found	Tagname	Ring	PCU	Module	Block	PIDBlock	PVBlock	PIDKind	File
☐	30FIC026	1	15	3	380	0	0	0	
☑	30FIC030	1	14	3	150	4535	4524	156	11403.cfg
☑	30FIC031	1	14	3	60	2040	2029	156	11403.cfg
☑	30FIC032	1	14	3	110	2180	2169	156	11403.cfg
☑	30FIC033	1	14	3	80	4635	4624	156	11403.cfg
☑	30FIC035	1	14	3	130	3255	3244	156	11403.cfg
☐	30FIC038	1	14	5	320	0	0	0	
☐	30FIC040	1	14	5	110	0	0	0	
☑	30FIC042	1	14	3	290	2280	2269	156	11403.cfg
☑	30FIC043	1	14	3	190	3038	3036	156	11403.cfg
☑	30FIC045	1	14	3	200	2340	2329	156	11403.cfg
☑	30FIC047	1	14	3	260	3445	3434	156	11403.cfg
☑	30FIC048	1	14	3	220	2440	2429	156	11403.cfg

Figure 215 - Loops detail table

The example above presents a list of loops coming from a large Infi90 project; since the loop scan was executed with a few module configuration files missing, only a part of the loops are actually

ABB. All rights reserved

found in the cfg files. These are marked as found loops and have their details filled in completely; missing loops have their details left partially empty.

Once the scan is complete, export filters may be implemented. This is accomplished in the second tab of the main form.

An area rule allows users to define a prefix able to distinguish between areas. For example, all tags beginning with 30 can be marked as belonging to area 3 by configuring an Area rule whose tag prefix is “30” and whose area is “Area 3”.

A new rule can be added by clicking with the mouse right button on top of the area rules table and by selecting **Add.** Once this is done, a new row is added in the table, with default values; both fields in the row cannot be changed according to user needs.

Any row can be erased using the **Delete** entry in the menu except the first one, that represents the default area for all loops. The **Clear** menu item allow deletion of all area rules but the default value.

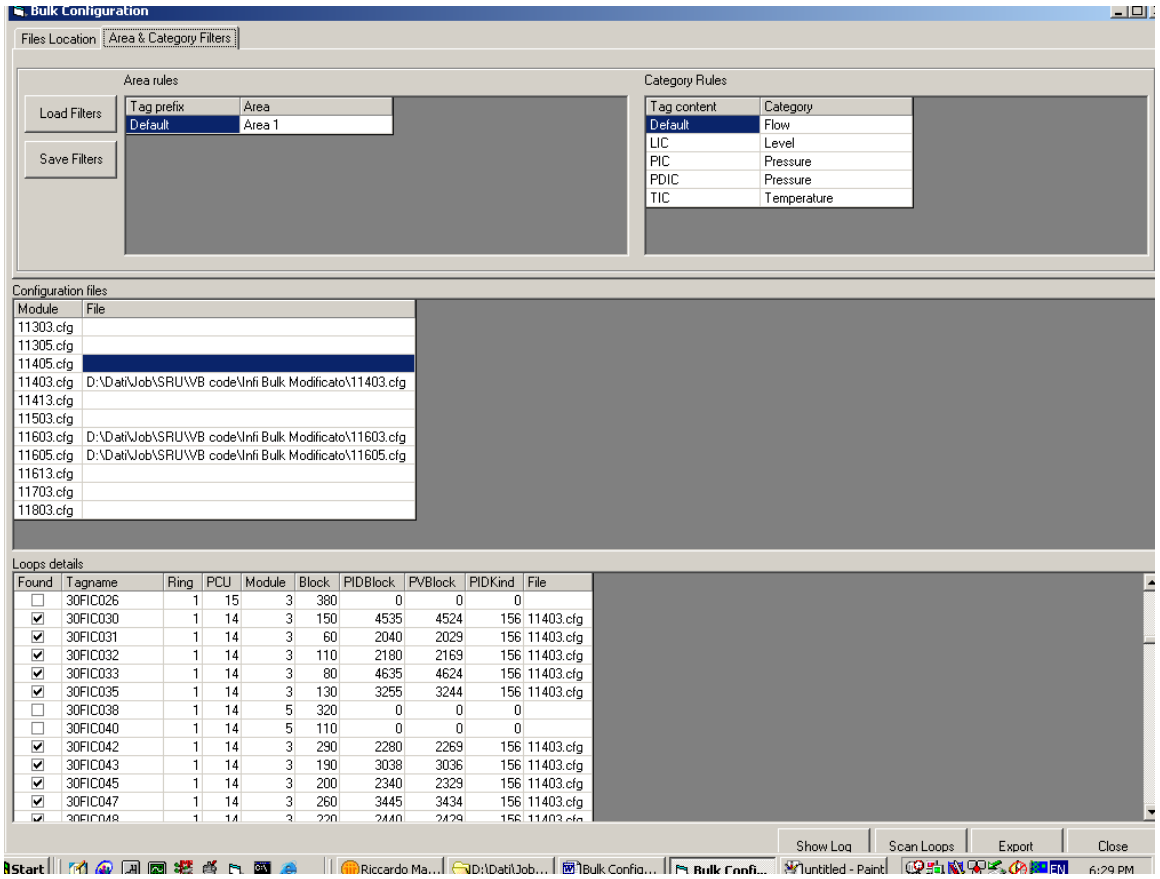


Figure 216 - Area and category rules

ABB. All rights reserved

16. Appendix H: Registry Parameter Settings

16.1 *Overview*

This appendix describes Windows Registry Keys that can be used to customize cpmPlus LPM software.

This section is intended for experienced users only. Modifications to LPM windows registry keys should be avoided and, where mandatory to provide the expected functionality, should always be performed with care and attention.

Improper modifications to LPM key may result in wrong LPM behavior or may even cause LPM to stop working. Resolution of issues related to improper key settings is always difficult for standard users.

16.2 *LPM Keys Path*

LPM Registry settings are saved in the following path

[HKEY_LOCAL_MACHINE\SOFTWARE\ABB\LPM\Configuration]

There is a single set of keys, accessed by LPM regardless of user and project configuration. All LPM keys except those ones dedicated to PDA features are saved into this location. PDA keys are saved into the path

[HKEY_LOCAL_MACHINE\SOFTWARE\ABB\LPM\PDA]

The next paragraphs provide information on LPM keys. Refer to the PDA manuals for further information on PDA keys.

ABB. All rights reserved

16.3 *LPM Keys details*

LPM main registry keys and their meaning/valid range are provided in the next table. The table might not contain all LPM keys but rather provides information on keys that may be modified by an experienced user.

The field Managed by HMI presents information on which keys are customized via use of the LPM Client – typically by modifying *Tool/Options* parameters. It is therefore not necessary to access directly to the registry to modify settings linked to these entries.

Key	Description	Internal Type	Applies to	Managed by HMI	Default
lNumFiles	Days of history storage [days]	Integer, 0-20	LPM Server	No	10
LpmK	Validation key.	String		Yes	
LPMTraceLevel	Trace level	Integer, 0-20	LPM Client	No	0 (=min trace)
MaxStrLenght	Maximum String length for object names (e.g. Areas, Loops, etc.)	Integer	LPM Client	No	30
MinStrLenght	Min length for object names	Integer	LPM Client	No	3
SaveRepinHTML	Save reports also in HTML	Boolean	LPM Server	Yes	1 (yes)

Table 16 – Registry Entries for LPM

ABB. All rights reserved